DIGITAL COMMUNICATION SYSTEM SIMULATION USING LabVIEW AS A SIMULATION TOOL

by

LT N NANGIA

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
January 2000

DIGITAL COMMUNICATION SYSTEM SIMULATION USING LabVIEW AS A SIMULATION TOOL

A thesis submitted
in Partial Fulfillment of the Requirements
for the degree of

MASTER OF TECHNOLOGY

by

LT N NANGIA

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
January 2000

CERTIFICATE

It is certified that the work contained in the thesis titled Digital Communication System simulation using LabVIEW as a simulation tool by Lieutenant Nitin Nangia Roll No 9810431 has been carried out under our joint supervision and this work has not been submitted elsewhere for a degree

(P K Chatterjee)
Professor
Department of Electrical Engineering
I I T Kanpur 208016

(J John)
Associate Professor
Department of Electrical Engineering
I I T Kanpur 208016

10th January 2000

CERTIFICATE

It is certified that the work contained in the thesis titled Digital Communication System simulation using LabVIEW as a simulation tool by Lieutenant Nitin Nangia Roll No 9810431 has been carried out under our joint supervision and this work has not been submitted elsewhere for a degree

(P K Chatterjee)	(J John)
Professor	Associate Professor
Department of Electrical Engineering	Department of Electrical Engineering
I I T Kanpur 208016	I I T Kanpur 208016

10th January 2000

ACKNOWLEDGEMENTS

I am grateful to Dr P K Chatterjee and Dr J John for having given me the opportunity to work under them I have benefited immensely from their guidance discussions and the literature provided by them throughout the progress of the thesis I would also like to express my deep and sincere gratitude to Dr P K Chaterjee for the constant guidance and the invaluable time given by him over useful discussions

Next I would like to thank Dr S Gupta and T K Agarwal for having helped me to understand the basics of programming in LabVIEW

I would also like to thank Maj V Mehta and V Sharmila for their encouragement and maintenance of a congenial environment The guidance and help from Mr A Warsi is also appreciated

Lastly I would like to express my deep appreciation to my wife Lucky without whose inspiration I could not have completed the work satisfactorily

(Nitin Nangia)

Place IIT Kanpur

Date 10 Jan 2000

ABSTRACT

Simulation model of various modulation and demodulation schemes used in digital communication systems has been implemented using a graphical programming software LabVIEW Data generators line coders modulators AWGN channel detectors and decoders have been modeled for all the modulation schemes simulated A signal from a function generator implemented in LabVIEW is the input source signal which has been processed and reconstructed

The effect of channel characteristics in terms of intersymbol interference and probability of error has been highlighted through simulation of zero forcing preset and adaptive equalizers It has been highlighted as to how the Inter symbol interference generated due to channel characteristics reduce noise margins and lead to higher errors in detection

The probability of error plots obtained for various modulation schemes the spectra, parameters determined over the link and the results of comparison of various modulated carrier signals in terms of power spectra phase transitions or transmission through a bandlimited channel match well with the theoretical results

CONTENTS

Chapter 4

RESULTS EVALUATION OF SIMULATION PERFORMANCE

LIST OF FIGURES

LIST OF TABLES

LIST OF PLOTS

CHAPTER 1

INTRODUCTION

1 1 LABVIEW & TOOLS OF ANAYLYSIS

Simulation may be defined as the discipline whose objective is to imitate one or more aspects of reality in a way that is as close to that reality as possible In a simulation based approach a computer is used to simulate the waveform or signals that flow through the systems Simulation of a communication system involves generating sampled values of signals and noise processing these sampled values through discrete time models of functional blocks in communication systems and estimating performance measures

A modular approach towards a communication system would show that it comprises of blocks which take input from preceding blocks process the signal and give their output for further processing These blocks could be a source coder modulator, channel filters detectors or decoders If these different blocks could be modeled separately and somehow combined to represent a complete communication system then it could be classified as a complete simulation The performance measure of the communication link simulation is the bit error rate obtained for various modulation / demodulation schemes in a AWGN simulation channel For baseband schemes the Eye diagram is taken as a subjective measure of performance

Digital simulation can provide a useful and effective adjunct to analytical performance evaluation or direct hardware evaluation of communication systems It is a computationally demanding task due to the large number of repetitive signal processing operations that must be performed in order to obtain a statistically valid measure of system performance Various computer languages have been used in the past like Simula, FORTRAN Pascal to simulate communication systems, using WICS or SKY VORTEX processors Due to the technological advancements, today a personal computer has the capability of such analysis and simulation using suitable software packages The software package used here for simulation is LabVIEW (Laboratory Virtual Instrument Engineering Workbench) LabVIEW is a program

development application much like C or BASIC having one basic difference that other programming systems use text based languages to create lines of code while LabVIEW uses a graphical programming language G to create programs in block diagram form

The main aim of the thesis has been to build independent menu driven blocks where the user can decide the nature of the communication link he wants to simulate by selecting appropriate modules At each stage the processed output is available for viewing graphically so as to understand better how the waveforms behave at various stages of a link The complete Passband and Baseband simulations can be run from two Panels shown in Figure 1 1 Each control knob when initiated gives further options for the user to select from

The **AIM** of the thesis can be defined as

1 To convert block concept level communication link into a modular menu driven utility which is user interactive
2 To construct modules of various modulation demodulation schemes
3 To simulate channel equalization and its effects
4 To assess performance of these detection schemes using the BER plots
5 To give the user a stage wise graphical representation of processing over the link

To understand the advantages of LabVIEW let us first understand how the programming is done There are three basic elements necessary to build any source code (i) Terminals (ii) Nodes and (iii) Wires The input parameters are operated upon by the function chosen and the output of this function block is the desired result The source code which is the Block diagram is hidden from the user and he has access only to the front panel where he can specify the inputs and see the results The following can be classified as the main advantages of this software package

(1) **Memory Usage** One of the main challenges of conventional language is the allocation and de allocation of memory LabVIEW a data flow paradigm of C removes this difficulty as you do not need to allocate variables nor allot values to and from them Instead you can create a diagram with connections representing the

transition of data Functions that generate data take care of allocating the storage for the data When data is no longer used the associated memory is deallocated

(2) **<u>Debugging Techniques & Performance Profiling</u>** LabVIEW has strong debugging tools like probes (to see values of data at intermediate points) breakpoints step running of program, highlight option which runs a VI in a slow mode highlighting all data values An Error VI indicates the various programming errors due to which the program is nonfunctional It also has performance profiling which is an interactive tabular display to view the performance data of sub VIs when called from a specific VI

(3) **<u>Graphical Displays and Data Acquisition</u>** LabVIEW (version 5 0 onwards) provides six types of graphs and two types of chart displays These graphs include the 3D surface graphs parametric graphs XY plots waveform graphs and waveform charts Data at any intermediate point in the link could be easily given for plotting Seeing output parameters visually without any complex source coding is a big asset associated with LabVIEW System parameters can easily be analyzed at intermediate steps if they are graphically represented

Data Acquisition (DAQ) card facilitates a high level interface to DAQ devices and signal conditioning hardware NI DAQ driver software interfaces the LabVIEW VIs with DAQ hardware devices and helps in developing complete instrumentation acquisition and control application Analogue or digital signals can be given as input parameters to LabVIEW for processing and also the output parameters through the interface to hardware devices Real time signals can be processed and analyzed using these DAQ interfaces

Another advantage of LabVIEW over other text based languages is that multiple processes can be initiated at one time This feature not only helps in synchronization but also helps in reducing the total processor time to complete a computation

(4) **<u>Frequently Used Functions</u>**

(a) **String to decimal** String function used in line code implementations for searching certain patterns of bit sequences

(b) **Index Array** If the output vector is one dimensional index array gives the parameter value at specified index

(c) **Subset Array** Returns a portion of the input vector

(d) **Sine wave** Generates sine waves of specified amplitude phase and frequency

(e) **Integral X(t)** Performs the discrete integration

$$F(t)=\int f(t)dt$$

$$\text{Integral } x = y_i = \frac{1}{6}\sum_{j=0}(x_{j-1} + 4x_j + x_{j+1})dt$$

(f) **Convolution** $h(t)=x(t)*y(t)$ where * stands for convolution

$$= \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau$$

$$h_i = \sum_{k=0}^{n-1} x_k y_{i-k} \qquad i=0\ 1\ 2 \quad n\ 1$$

(g) **Cross correlation** $R_{xy}(t) = x(t)*y(t) = \int_{-\infty}^{\infty} x(\tau)y(t+\tau)d\tau$

where * stands for convolution

$$h_j = \sum_{k=0}^{n-1} x_k y_{j+k}$$

$$Rxy_i = h_{i\ (n\ 1)} \qquad \text{for } i=0\ 1\ 2 \quad (n-1)$$

(h) **FFT** Computes real FFT or real DFT of input sequence

(i) **Amplitude and Phase spectrum** Computes single sided scaled amplitude spectrum magnitude and phase of a real time domain signal

(j) **Butterworth filter** Can be used as a low pass/band pass filter Sampling frequency low and high cut off frequencies and the order have to be specified

(k) **erfc(x)** $\frac{2}{\sqrt{x}}\int_{x}^{\infty} \exp(-t^2)dt$

(l) **Standard deviation** $\sigma_x = \sqrt{\frac{1}{n}\sum_{=0}^{1}(x - y)^2}$

where $y = \frac{1}{n}\sum_{i=0}^{n-1} x_i$ n = number of elements

(m) **Spline interpolation** used for interpolating discrete points for graphical representation

(n) **Solve linear equations** solves a real linear system AX=Y where A X and Y are vectors

(o) **Create special matrix** To generate nx by ny toeplitz matrix

(p) **Analogue Input acquire waveforms** used in DAQ program s Acquires data from the specified channels and samples the channels at the scan rate

1 2 GENERAL DESCRIPTION OF COMMUNICATION SYSTEMS

The schematic of a basic communication system shown in Figure 1 0 have been coded using LabVIEW Detailed and software implementation are illustrated in the subsequent chapters A brief review of some basic concepts are appended below

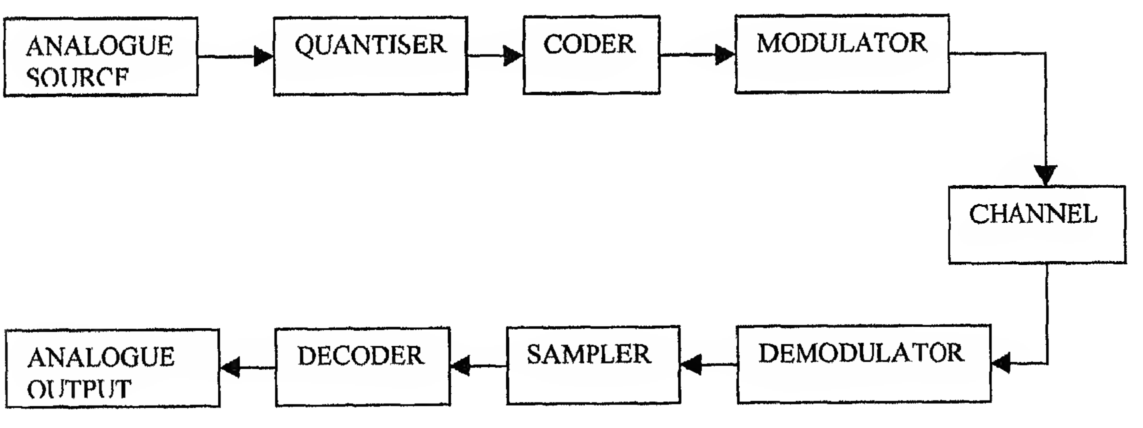


Figure 1 0 Basic communication System Block

Sampling

The sampling theorem states that a band limited signal having no spectral components above f_m hertz can be determined uniquely by values sampled at uniform intervals of T_s seconds where

$$T_s \leq 1/(2f_m)$$

The upper limit on T_s can be expressed in terms of the sampling rate denoted

$$f_s = 1/T_s$$

$$f_s \geq 2f_m$$

The sampling rate $f_s = 2f_m$ is also called the Nyquist rate [10]

Analogue to Digital conversion

The process of digitizing the original analogue signal is called **Quantisation** It consists of mapping the amplitudes of the signal into a number of discrete amplitude levels This process results in loss of information since it is not possible to recover the original analogue signal [11]

Orthogonality Principle

Signals that satisfy the following condition are orthogonal to one another

$$\int_0^T s_i(t) s_j(t) dt = E\delta_{ij}$$

where $\delta_{ij} = 1 \qquad i \neq j$

$\qquad = 0 \qquad i = j$

An N dimensional orthogonal space can be defined as a set of N linearly independent functions $\phi_j(t)$ called the basis function Each $\phi_j(t)$ function of the set of basis function to satisfy the principal requirement for orthogonality must be independent of the other members of the set and it must not interfere with the detection process of the other members of the set [13]

Probability of Error in AWGN Channel

$s(t) = A\cos\omega t$

$\quad = \sqrt{2P}\cos\omega t$

$$= \sqrt{\frac{2E}{T}} \cos \omega t$$

where P represents the average Power in watts and E the energy per bit [22]

$$\frac{E_b}{N_0} = \frac{ST}{N_0} = \frac{S}{RN_0} = \frac{SW}{RN_0W} = \frac{S}{N}\left(\frac{W}{R}\right)$$

Eb = signal energy per bit
No = noise spectral density
S = signal power
T = bit duration
R = Rate
W = bandwidth
A = Amplitude
ω = 2π(frequency)

Correlation Receivers

A matched filter and a product integrator both perform the same task

$$Z(T) = \int_0^T r(\tau)s(\tau)d\tau$$ where τ is the time difference

The product integration of the received signal r(t) with a replica of the transmitted signal s(t) is known as correlation of r(t) with s(t) At time t = T the output of the correlator and matched filter is identical

Response of Correlators to White noise

$$\sigma_j^2 = \frac{N_0}{2}$$

where No/2 is the power spectral density and σ_j^2 is the noise variance [22]

ISI and Equalization

A band limited channel spreads a pulse waveform and causes pulse overlap This overlapping of pulses is called Intersymbol interference (ISI) An equalizer filter is used to compensate for the distortion caused due to the channel The minimum bandwidth needed to detect R_s symbols/sec without ISI is $R_s/2$ Hz If the rate increases beyond 2B where B is the bandwidth it effects the discrete samples given to the sampler for detection ISI reduces the noise margins and can lead to higher errors

The process of correcting the channel distortion is called equalization A (2N+1) tap transversal filter with optimal tap coefficients determined by the peak distortion or mean square distortion criterion forces the equalizer output to zero at N points on either side of the desired peak The weights in MSE multiplied with the equalizer input adds up to the polar response of the symbol (1 or –1) being sent In both cases N simultaneous equations are solved [21]

1 3 PROBLEM FORMULATION

The thesis involved simulation of the generic communication system model using LabVIEW package and the evaluation of system performance with various modulation demodulation schemes The thesis covers the following

(I) Passband Systems

(a) **Generation** of an analogue source
(i) Generation of sinusoid signal at various frequencies amplitude and phase
(ii) Generation of random signal
(iii) Signal taken from function generator using DAQ utility

(b) **Conversion** of analogue signal to digital using linear and non linear quantisation techniques
(i) 4 level linear quantiser
(ii) 7 level linear quantiser
(iii) A law non linear quantiser
(iv) μ law non linear quantiser

(c) **Lines codes**
(i) Alternate Mark Inversion coder (AMI)
(ii) m Binary n Binary coder (mB nB)
(iii) Bipolar Six Zero Substitution coder (B6ZS)
(iv) Biphase coder
(v) 4 Binary 3 Ternary coder (4B3T)

(d) Simulation of various digital **modulation** schemes
(i) Amplitude shift keying (ASK)
(ii) Coherent and non coherent frequency shift keying (CFSK & NCFSK)
(iii) Binary phase shift keying (BPSK)
(iv) Quardrature phase shift keying (QPSK)
(v) Minimum shift keying (MSK)
(vi) 16 QAM and 64 QAM
(vii) 3 PSK and 3 FSK
(viii) Offset quardrature phase shift keying (OQPSK)

(e) Simulation and generation of **AWGN** samples

(f) Simulation of **demodulators** (of all modulators)

(g) Reconstruction of input signal after **decoders** (using different coders and quantisers)

(h) Evaluating system **performance** for different SNR in terms of probability of error

(i) Determining spectrums and parameters over the link

(j) Evaluation of the effect of bandpass filters on QPSK and OQPSK signals

(II) Baseband Systems

(a) **Generation** of binary source sequence

(b) Performance of **zero forcing equalizer** using different channel impulse response

(c) Performance of a preset equalizer and decision directed linear equalizers

(d) **BER** evaluation of decision directed linear equalizer

(e) **Channel tracking** and errors in decision directed linear equalizer

(f) Generation of **Eye diagrams** before and after equalization

The above operations include all functions of transmitters AWGN channel and receivers A known bit sequence is transmitted duly corrupted and received The comparison in terms of probability of error of the transmitted and the received bits gives the system performance

1 4 Organization of Thesis

After the introductory chapter dealing with advantages of LabVIEW, basic description of communication system model and underlying principles Chapter 2 explains the various Passband modulation schemes data generation line coding quantisation, decoders and demodulators Chapter 3 deals with Baseband schemes and Equalization Chapter 4 brings out the various results like the probability of error in various modulation schemes errors caused due to ISI errors caused due to bandpass

filtering eye diagrams and certain parameters determined over the link Chapter 5 deals with the concluding remarks and suggestions for future work

Some of the important programs used in the study are appended in the Appendix The full package consisting of 300 files (65 MB) is submitted on a CD along with the thesis report

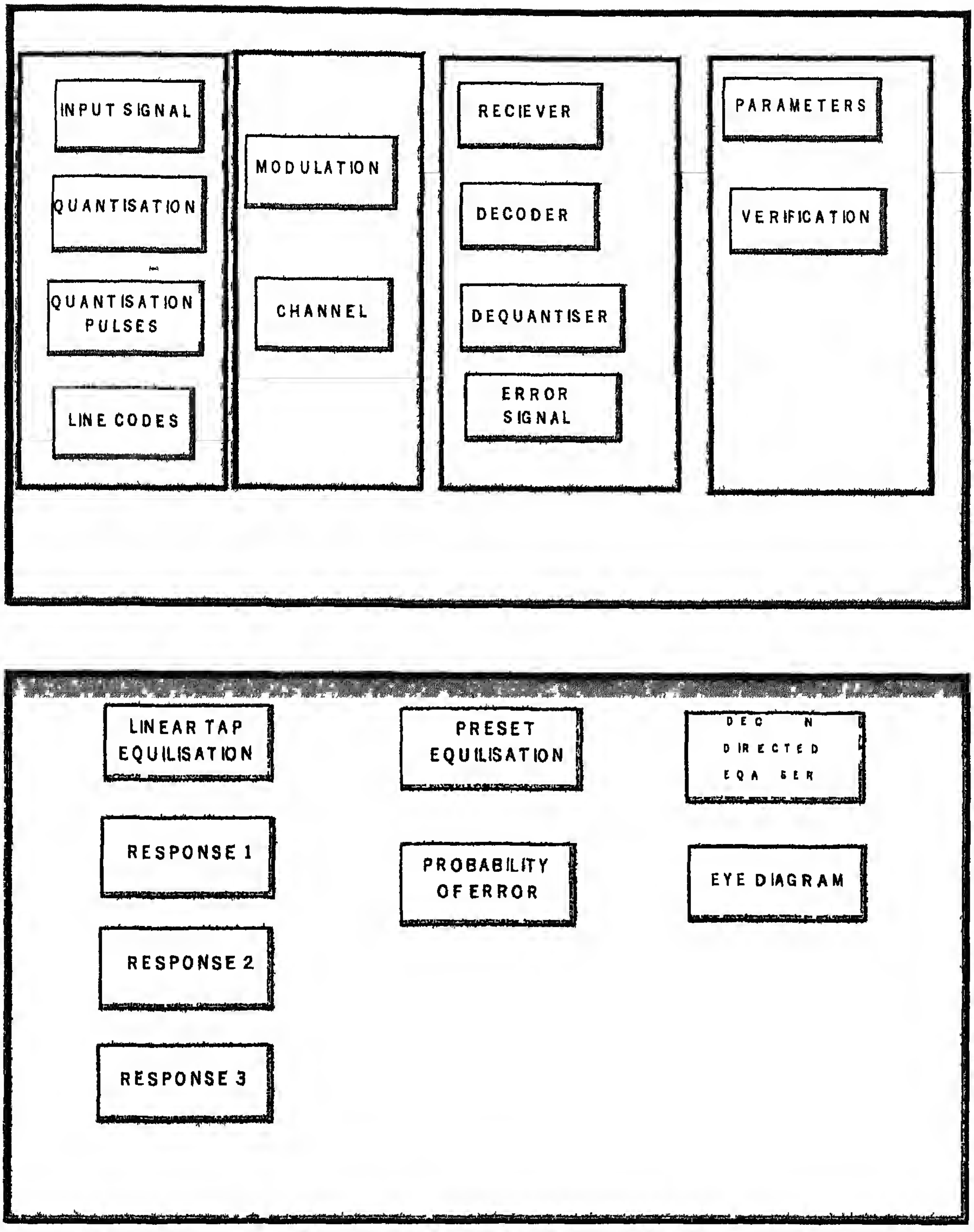


Figure 1 1   FRONT PANEL SELECTION

CHAPTER 2

PASSBAND MODULATION AND DEMODULATION SCHEMES

2 1 0 DATA GENERATION AND CODING

The generation of analogue waveform is simulated in three different ways

(i) Sinusoidal waveform generation using a function generator using function vi

(ii) Taking an analogue input signal from a function generator (external source) using Analogue Input vi

(iii) Random signal generation using random vi

Figure 2 1 shows the front panel of the function generator The parameters available to the users are the frequency, phase and amplitude The generated waveform (sine sawtooth or square) can be visually seen on the graphical display The frequency is defined by the sampling rate specified Upto $f_s/2$ Hz of frequencies can be generated on the function generator where f_s is the sampling frequency The frequency in cycles/sec divided by the sampling rate gives the normalized frequency[2]

$$\frac{\text{cycles / s}}{\text{samples / s}} = \frac{\text{cycles}}{\text{sample}}$$

The reciprocal of the normalized frequency gives the no of samples required to generate one cycle of the sine wave By controlling the fine and course knobs we can generate frequencies upto $f_s/2$ A digital display facilitates the user to see what frequency has been generated much like checking the frequency on a CRO The amplitude and phase can also be adjusted DC offset or noise can also be added to the generated waveform

The random number generator function is utilized to generate a random waveform pattern (uniformly distributed) The user specifies the range that is the maximum and minimum values of the voltage

The third option is to take in a signal from an external source a function generator, through the DAQ interface The scan rate specified in the program specifies upto to what value the frequencies can be generated The scan rate is a variable parameter Acquire waveform vi using the intermediate level DAQ vis is used to acquire waveforms for further processing The select input vi is a menu driven utility which selects the desired input from the above options

2 1 1 Quantisation

The conversion of simulated analogue waveform to discrete values is done via the quantisation vi These are classified into four quantisation processes

(1) 16 level linear quantisation

(2) 128 Level linear quantisation

(3) A law nonlinear quantisation

(4) u law nonlinear quantisation

Implementation of 128 level linear quantisation program (Program #1)

1 Determine the range (maximum voltage minimum voltage)

2 Divide range by 128

3 Run iteration $x2^n$ 7 times giving output (2^6 2^5, 2^4 2^3 2^2 2^1 2^0)

4 Multiply the above array o/p with the o/p of (range/128) We get the X*Y array

5 Classify input as + or and give input to appropriate case statements

6 Compare first value of X*Y array with the input, if the input value is greater, subtract the two and now compare to the second value of the X*Y array Assign Boolean True if value is smaller, or False if greater Use shift register to store added values

7 Carry out above operation for all 7 bits and get an array of Boolean true and false

8 Convert Boolean to 0,1 (1= true 0= false)

9 Each input voltage sample is converted to its corresponding seven bits

10 For negative input carry out above conversion to Boolean in the exact manner described using the case statement for the –ve inputs

The above is the 2 s compliment method of Quantisation To illustrate numerically let the input voltage value be +4V (The peak input voltage is +5V and the number of levels is 16)

10/16=0 625

[X*Y]= [5   2 5   1 25   0 625]

Compare 4 with 5 0 assign False	is 4 > 5
Compare 4 with 2 5 assign True	is 4 > 2 5
Compare (4 2 5) with 1 25 assign True	is 1 5 >1 25
Compare (4 2 5 1 25) with 0 625 assign False	is 0 25 >0 625

So we get for an input voltage of 4V a binary of 0110 (Refer to Table 2 1) For some range of input voltage values a binary number is assigned This range of input voltage is called the step size $[q = \frac{P}{M} = \frac{2V}{M}]$ Here V=5 volts and M is 16 The voltage values are assigned in steps of 0 625V[10]

NonLinear Quantisation

Tapering of the quantisation level leads to better performance as compared to the equal spacing quantisation The input signal is non linearly compressed in amplitude to force all signals to lie within a specified range This compression is typically of logarithmic form The μ Law and A Law are two such classifications [16]

μ Law

$$y(x) = \frac{\ln(1+\mu|x|/V)}{\ln(1+\mu)}\operatorname{sgn}(x) \qquad |x| \le 1$$

A law

$$= \frac{Ax}{1+\ln A} \qquad |x| \le \frac{1}{A}$$

$$y(x) = \frac{1+\ln A|x|}{1+\ln A}\operatorname{sgn}(x) \qquad \frac{1}{A} \le |x| \le 1$$

Typical value of A being 100 and μ is 255 Where y(x) is defined as the quantised level assigned for x the normalized input signal

VI for μ law quantisation is described below

1 Sign magnitude format is used where the first bit gives the polarity The remaining 7 give the magnitude (Refer to **Table 2 2**)

2 Assign 1 if polarity is positive Assign 0 if polarity is negative

3 Next find S the segment identifier If input voltage between the range 0 to 31 it is mapped to segment 1

$x<64\ 2^{a}\ 33$ $\qquad$ x = sample value

$\qquad$ a= 0 1 7

4 Next determine **residue** r that is the difference between input amplitude and the lower end point of the segment

$r = x$ $\qquad$ $s = 0$

$r = x\ (32\ 2^{s}\ 33)$ $\qquad$ $s = 1\ 2\ \ 7$

5 Next determine b the **quantisation interval** containing r

$r < 2b+1$

$r < (2^{s+1})(b+1)$ $\qquad$ s=1 2 7

6 On getting polarity S and b assemble the 8 bits Sample input has been mapped to binary 8 bits

7 Invert bits for transmission The range of encoding Table is upto 8159 If input voltage value exceed this limit an error signal is generated Illustrating the above numerically

Input voltage = 24 volts

+ve $\qquad$ Assign $\qquad$ 1

Segment 1	Assign	000
b is 12	Assign	1100
Binary	**10001100**	

A law follows similar algorithm [16]

2 1 2 LINE CODE

Coding is the transformation of the source bits to the transmitted data symbol The coders add redundancy to the transmitted data symbols and at the receiver end checks are made to see wheather these code constraints are preserved or not Some of the advantages of coding are

1 To place spectral nulls at dc thereby enabling baseband transmission through a c coupled channels
2 To add redundancy and detect errors
3 Line codes allow concentration of signal power at frequencies near dc
4 Line codes have effective control over the power spectrum
5 Line codes assist in extracting timing or clock information from the signal

The following codes have been implemented which form the coding module These menu driven options are

1 Biphase coder
2 Alternate Mark Inversion coder (AMI)
3 Bipolar Six Zero Substitution coder (B6ZS)
4 m Binary n Binary coder (mBnB)
5 4 Binary 3 Ternary coder(4B3T)

Of these mBnB coding and its implementation is discussed below

mBnB coder

DISPARITY AND DIGITAL SUM

The disparity of a digit (0 1,2 r 1) is defined as[15]

$$d_i = a_i \quad \overline{a_1} \qquad \text{(I)}$$

where a_i is the ith digit, $\overline{a_1}$ is the complement of it, and r is the radix

$$\overline{a_1} = (\mathrm{r}\ 1) - a_\mathrm{i} \qquad \text{(II)}$$

The disparity of an n digit code word is given by

$$d = \sum_{1=1}^{\mathrm{n}} \mathrm{d}_1 = 2\sum_{1=1}^{\mathrm{n}} a_I - \mathrm{n}(\mathrm{r}-1)$$

$$= 2\sum_{i=1}^{n} a_i \quad \mathrm{n} \quad \text{(for binary transmission)}$$

$$= \text{(No of 1 s)} - \text{(No of 0 s)}$$

The digital sum d_s, of an n digit code word is given by

$$d_s = \sum_{i=1}^{n} a_i$$

$$a_i = \begin{cases} -0\ 5 \text{ for a zero} \\ \ \ 0\ 5 \text{ for a one} \end{cases}$$

In the mBnB code the binary sequence which is to be transmitted is first broken into m bits and then converted to n bits and transmitted (where n > m) The number of combinations of each code having a particular disparity (assuming n = m +1 and m is odd) is given by

No of words having zero disparity $= {}^{m+1}C_{(m+1)/2}$

No of words having Non zero disparity $= 2^{m+1}\ {}^{m+1}C_{(m+1)/2}$

A source word is translated to zero disparity or a pair of words having alternate disparity

3B4B code

3 bit input data is converted to 4 bit output data

RDS max = 1

RDS min = 1

where RDS (Running digital sum) at the k^{th} bit is defined as

$$\mathrm{RDS}(\mathrm{k}) = \sum_{1=-\infty}^{\mathrm{k}} a_I$$

DSV(digital sum variation) = RDSmax – RDSmin

= 2

the coding done should give the minimum DSV

IMPLEMENTATION (Program # 2)

1 Initiate all input variables to initial value

2 Reshape input array from 2 dimensional to one dimensional array of 3 bit serial input using the reshape array function

3 Refer to **Table 2 3** Map 3 bit serial input directly to 4 bit binary codeword if the code and its complimentary are same

4 If not same, calculate disparity If DSV > 2 then select ve disparity code and if disparity < 2 then select +ve disparity code This is implemented by using While Loop for disparity calculation and then using appropriate case statements that do the 3B to 4B mapping

5 Repeat step 2 through 4 for all input bits and reconstruct the mapped one dimensional array of the converted 4 bits

6 Give this output to chart for plotting

BIPOLAR SIX ZERO SUBSTITUTION (B6ZS)

To modify the linear AMI line code to a nonlinear code, B6ZS block line code is used This class of code modifies the AMI code by performing a substitution for a block of six consecutive zeros For each block of six zeros the code word B0VB0V is transmitted where B stands for binary and V for violation If the RDS at the start of the block is zero then, +0+ 0 0 would be substituted for the six zeros, otherwise –0 +0+

4 BINARY 3 TERNARY

In this block code binary words are mapped into ternary words, with the use of feedback to choose the appropriate mode at each block so as to minimize the RDS 4B3T is a bimode code where mode A is chosen whenever the RDS at the beginning of the 4 bit block is in range 3≤RDS≤ 1 and mode B is chosen when the RDS is in range 0≤RDS≤2 Table 2 4 has been used to implement 4B3T coder

2 2 MODULATION SCHEMES

The following modulation schemes have been incorporated using LabVIEW A brief theoretical preview will follow the actual implementation

1 Amplitude shift keying (ASK)

2 Coherent and Noncoherent frequency shift keying (FSK& NCFSK)

3 Binary phase shift keying (BPSK)

4 Quararature phase shift keying (QPSK)

5 Minimum shift keying (MSK)

6 16 Quardrature Amplitude Modulation (16 QAM)

7 64 QAM

8 3 Phase Shift Keying (PSK)

9 3 Frequency Shift Keying(FSK)

10 Offset Quardrature Phase Shift Keying (OQPSK)

The **parameters** used in the theoretical expressions for the following sections are defined as

E_b = Signal energy per bit

T_b = Bit duration

ω = $2\pi f$ where f is frequency

$\phi(t)$ = Basis function

h = deviation ratio

A = Amplitude

$\theta(t)$ = phase

2 2 1 Binary Phase shift keying

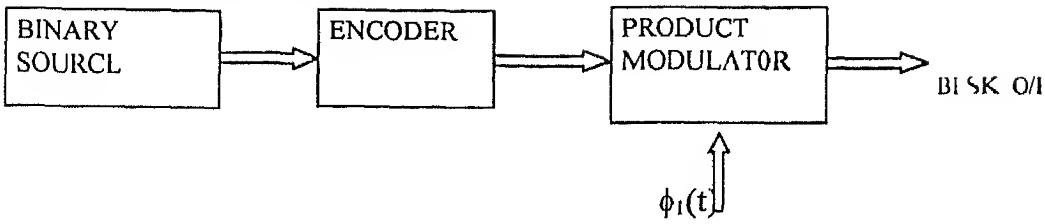


Fig 2 2 BPSK TRANSMITTER

The general analytical expression for PSK is [14]

$$s_i(t) = \sqrt{\frac{2E_b}{T_b}}\cos[\omega_0 t + \theta_i(t)] \qquad 0 \le t \le T_b$$

$$i = 0\ 1$$

$$\theta_i(t) = \frac{2\pi i}{M} \qquad M = 2$$
$$= \pi i$$

where $\phi_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t)$

The data signal shifts the phase of the waveform from one of the two states that is, either zero or π The signals are said to be antipodal (Refer Figure 2 2)

Implementation

1 Generate carriers that are offset by 180 degrees using the PSK carrier vi

2 Depending on the input bit, shift the phase of the carrier from 0 to 180 degrees

3 Plot output on chart Give output to global modulated O/P'

2 2 2 Non Coherent FSK

$$s_i(t) = \begin{cases} A\cos(2\pi f_i t) & 0 \le t \le T_b \\ 0 & elsewhere \end{cases}$$

f_i equals f_1 and f_2

$f_i = n_i/T_b$ where n_i is an integer and T_b is the bit duration The two frequencies are orthogonal (They satisfy the othogonality principle) f_1 represents a 0 and f_2 represents a binary 1 [11]

Implementation

1 Select f_c and using h = 1 (h is the deviation ratio)and R (Bit rate) generate f_1 and f_2

2 Using nfsk carr vi generate carriers at frequencies f_1 and f_2

3 Modulate carriers with the incoming bits and store in global "non coh o/p ' for further processing Give to chart to plot output

2 2 3 QUARDRIPHASE SHIFT KEYING

The information carried by the transmitted signal is carried in the phase[13]

$$s_i(t) = \begin{cases} A\cos(2\pi f_c t + (2i-1)\frac{\pi}{4}) & 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases}$$

$$s_i(t) = \sqrt{\frac{2E}{T}}\cos[(2i-1)\frac{\pi}{4}]\cos(2\pi fct) - \sqrt{\frac{2E}{T}}\sin[(2i-1)\frac{\pi}{4}]\sin(2\pi fct)$$

where $i = 1\ 2\ 3,4$ E = symbol energy T = symbol duration

Orthonormal basis functions are

$$\phi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi fct)$$

$$\phi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi fct)$$

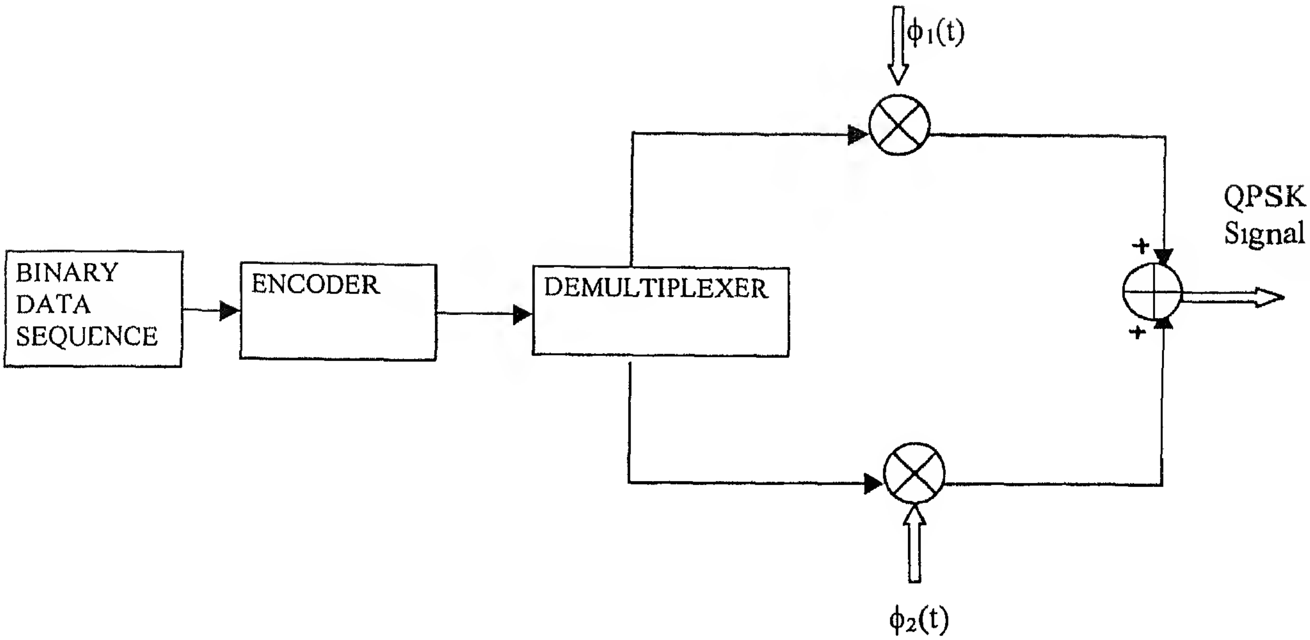


Fig 2 3 QPSK TRANSMITTER

Signal vectors associated with the four message points are

$$s_i(t) = \begin{bmatrix} \sqrt{E}\cos((2i-1)\frac{\pi}{4}) \\ -\sqrt{E}\sin((2i-1)\frac{\pi}{4}) \end{bmatrix} \quad \text{where } i = 1\ 2\ 3\ 4 \qquad \text{(I)}$$

Implementation

1 Initialize all input variables to default values

2 Generate ϕ_1 (t) and ϕ_2 (t) 90° offset from one another at fc

3 Generate signal space diagram by varying i in the above equation (I) and give to an XY graph for plotting

4 De interleave the input bits into odd and even bits

5 Generate ϕ_1 (t) or ϕ_2(t) over two bit interval (2T) for the Inphase and Quardrature channel

6 Add the Inphase and Quardrature channel signalse, the resultant is the QPSK signal waveform

7 Plot the Inphase, Quadrature and the resultant QPSK waveform for the complete set of incoming bits

2.2.4 MINIMUM SHIFT KEYING

In QPSK due to the alignment of the inphase and quardrature (a_1 (t) and a_2 (t)) streams the carrier phase can change over every 2T A change in both components simultaneously can result in a phase shift of 180° This phase shift in band limited operations results in the envelope going to zero The restoration of which leads to restoration of all frequency sidelobes back to their original level To circumvent this problem the modulation scheme minimum shift keying was developed , where the phase of the outgoing wave from the transmitter has no transitions it is continuous in phase[24]

$$s(t) = \sqrt{\frac{2Eb}{Tb}}[\cos(2\pi f_1 t + \theta(0))] \quad \text{for symbol } 1 \qquad 0 \le t \le T_b$$

$$s(t) = \sqrt{\frac{2Eb}{Tb}}[\cos(2\pi f_2 t + \theta(0))] \quad \text{for symbol } 0 \qquad 0 \le t \le T_b$$

where

$$\theta(t) = \theta(0) \pm \frac{\pi h}{Tb} t \qquad 0 \le t \le Tb$$

$$f_c + \frac{h}{2T_b} = f_1$$

$$f_c \quad \frac{h}{2T_b} = f_2$$

or

$$f_c = \frac{1}{2}(f_1 + f_2)$$

$$h = T_b(f_1 - f_2)$$

where h is the deviation ratio (h = ½ for MSK)

The orthonormal basis functions φ_1 (t) and φ_2 (t) are defined as the sinusoidally modulated quardrature carriers

$$\varphi_1(t) = \frac{\sqrt{2}}{\sqrt{T_b}}\cos(\frac{\pi}{2T_b}t)\cos(2\pi fct) \qquad 0 \le t \le T_b$$

$$\varphi_2(t) = \frac{\sqrt{2}}{\sqrt{T_b}} \sin(\frac{\pi}{2T_b} t) \sin(2\pi fct) \qquad 0 \le t \le T_b$$

$s(t) = s_1 \varphi_1((t) + s_2(t) \varphi_2(t)$

Where $s_1 = \int_{T}^{T_b} s(t)\varphi_1(t)dt$

$s_2 = \int_{0}^{2Tb} s(t)\varphi_2(t)dt$

Signal Space Characterization of MSK

Transmitted Binary Symbol	Phase States		Coordinates	
	$\theta(0)$	$\theta(T_b)$	s_1	s_2
1	0	$+\pi/2$	$+\sqrt{E_b}$	$-\sqrt{E_b}$
0	π	$+\pi/2$	$\sqrt{E_b}$	$-\sqrt{E_b}$
1	π	$\pi/2$	$\sqrt{E_b}$	$+\sqrt{E_b}$
0	0	$\pi/2$	$+\sqrt{E_b}$	$+\sqrt{E_b}$

Table 2 5

The above table is used in selecting the phase and the polarity of the scaled time function $s_1 \varphi_1(t)$ and $s_2 \varphi_2(t)$ $s_1 \varphi_1(t)$ and $s_2 \varphi_2(t)$ are offset by T_b The MSK waveform is generated by adding these two waveforms This resultant waveform is continuous in phase

Implementation (Program # 3)

1 MSK parameter vi determines the f1 and f2 carrier frequencies The input parameters required are fc, R(rate) and h (h=1/2)

2 MSK trellis vi generates the phase trellis using the input bits A $\pi/2$ transition is defined for bit 1 and $\pi/2$ transition for a bit 0 The trellis is plotted on a XY graph

3 Generate MSK orthonormal basis functions $\varphi_1(t)$ and $\varphi_2(t)$ using the MSK carrier vi generating carriers at 180° offset for $\varphi_1(t)$ and $\varphi_2(t)$

4 De interleave the input bit stream into odd and even bits (Inphase and Quardrature)

5 Determine polarity of s_1 for the inphase bits using the signal space Table 2 5 using the incoming bit and $\theta(0)$

6 Determine polarity of s_2 for the quardrature bits using the incoming bit and $\theta(T_b)$

7 On getting the polarity (positive or negative) of s_1 and s_2 generate $s_1\varphi_1(t)$ and $s_2\ \varphi_2(t)$ both over $2T_b$, $s_2\ \varphi_2(t)$ being offset by T_b

8 Add the Inphase and Quradrature channel waveforms The resultant is a MSK waveform

Thus MSK modulation determines the polarity of the carrier by seeing the previous phase

The four possible cases can be

1	$\theta(0) = 0$	$\theta(T_b) = \pi/2$	BIT 1
2	$\theta(0) = \pi$	$\theta(T_b) = \pi/2$	BIT 0
3	$\theta(0) = \pi$	$\theta(T_b) = \ \pi/2$	BIT 1
4	$\theta(0) = 0$	$\theta(T_b) = \ \pi/2$	BIT 0

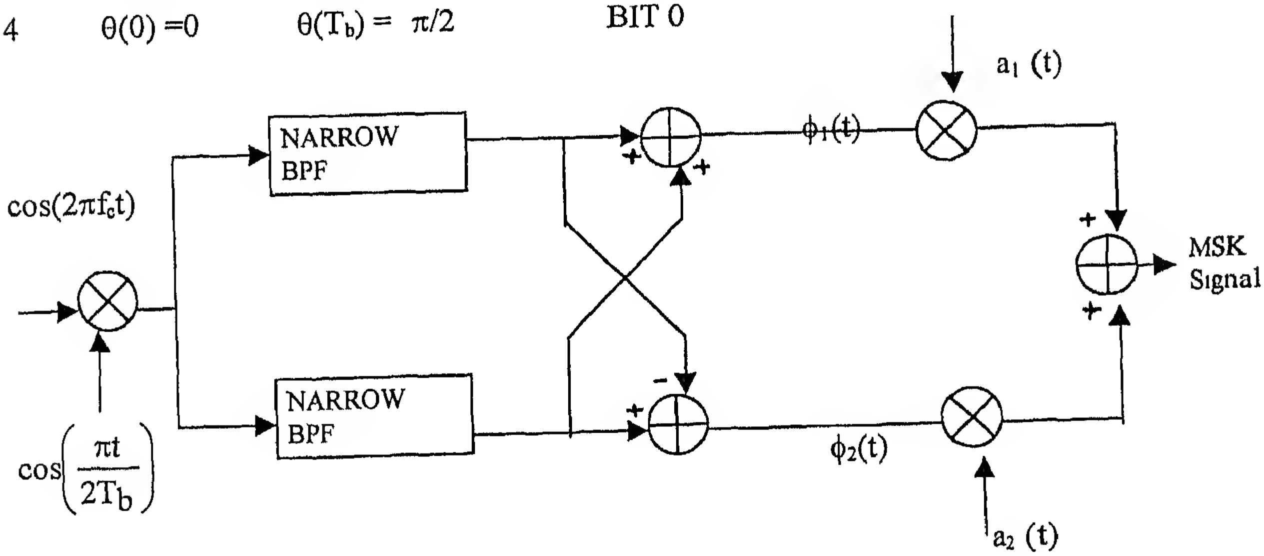


Fig 2 4 MSK TRANSMITTER

2 2 5 DIFFERENTIAL PSK

DPSK consists of two basic operations differential encoding of the binary input sequence and phase shift keying To send a symbol 0 phase advance the current signal waveform by 180 degrees and to send a symbol 1 we leave the phase unchanged The relative phase difference between two successive bit intervals is stored in DPSK[17]

To encode the incoming bit stream a reference symbol is the arbitrary first bit Let b_k be the incoming bit sequence then the differentially encoded bits are the complement of the module 2 summation of the b_k and d_{k-1} (Refer to **Figure2 5**) The differentially encoded bits modulate a carrier wave f_c to produce the DPSK signal

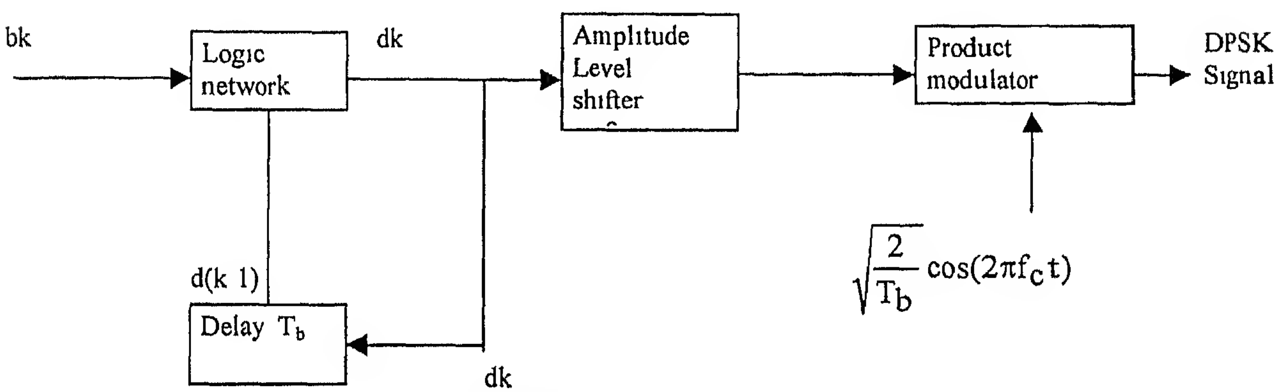


Fig 2 5 DPSK TRANSMITTER

Implementation

1 The initial reference bit is taken as 1

2 The input bit stream b_k is converted to Boolean and XOR operation is carried out with the preceding bit i e , d_{k-1} to generate d_k (refer Table 2 6)

3 DPSK carrier is generated using DPSK carr vi

b_k		1	0	0	1	0	0	1
d_{k-1}		1	1	0	1	1	0	1
d_k	1	1	0	1	1	0	1	1
Transmitted phase	0	0	π	0	0	π	0	0

Table 2 6

4 The encoded bits modulate the carrier i e a change from 1 to 0 the phase changes by π and a change from 0 to 1 the phase sent out is 0 For binary 1 the phase is 0 and for binary 0 it shifts to π

5 The DPSK waveform is plotted on chart and output is given to global variable modulated O/P '

2 2 6 64 QAM TRANSMITTER

In the M ary Quardrature amplitude modulation scheme the carrier experiences amplitude as well as phase modulation The general form of M ary QAM is defined by

$$s_i(t) = Aa_i \cos(2\pi f_c t) + Ab_i \sin(2\pi f_c t) \qquad 0 \le t \le T$$

$$a_i = \pm 1 \ \pm 3 \quad \sqrt{M} \ 1 \qquad b_i = \pm 1 \ \pm 3 \quad \sqrt{M} \ 1$$

where a_i and b_i are chosen in accordance with the location of the message point Each constellation point in the square constellation is represented by a unique 6-bit symbol which is Gray coded to minimize the error probability[19]

Implementation (program # 4)

1 Generate the square constellation signal space diagram for 64 QAM using space vi

2 6 bits which form a symbol which are given as input to the For Loop The For Loop does the following repetitive processing on all the 6 bit input arrays

3 The 6 bits are further split into two groups of 3 bits by the Array subset function

4 The decimal to string conversion is used to convert to strings

5 Refer to square 64 QAM phasor constellation diagram (Figure 2 6) Group I of 3 bits is mapped to the first 3 bits and Group II to the last 3 bits of the 6 bits assigned to the message points of the constellation These bits are gray coded

6 On doing the mapping and knowing where the message point is determine a_i and b_i which would be 1, 3 5 7 1 3 5 or 7 on the real and imaginary axis

7 Generate QAM carriers $A\cos(2\pi f_c t)$ and $A\sin(2\pi f_c t)$ using QAM carrier vi

8 Multiply a_i with $A\cos(2\pi f_c t)$ and b_i with $A\sin(2\pi f_c t)$ and add the two The resultant is the QAM waveform

9 Plot QAM waveform on chart and give modulation O/P to global variable modulated O/P

NOTE 16 QAM is also implemented in a similar manner

2 3 DEMODULATION TECHNIQUES

The next block considered after the transmitter is the receiver The channel is assumed to be an ideal AWGN Channel which adds gaussian noise to the transmitted signal We will break up the receiver into two parts First we will receive input back to back adding no noise Next we will see the performance of these receivers under varying AWGN variance This latter part will be covered under the result section The receivers of all modulation schemes discussed in section 2 2 are developed using either the correletoi detector or matched filter Amongst these illustration and implementation of the following receivers is given

1 Binary phase shift keying
2 Quardratuie phase shift keying
3 Minimum shift keying
4 64 Quardrature amplitude modulation
5 Non coherent frequency shift keying

(Proper carrier, phase and bit timing synchronization are assumed in the receivers for the various coherent schemes)

2 3 1 Binary Phase Shift Keying

A pair of signal s_1 (t) and s_2 (t) is used to represent a binary 1 and binary 0

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f c t)$$

$$s_2(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f c t)$$

The basis function

$$\varphi_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f c t)$$

The received signal is

$r(t) = s(t) + n(t)$ where s(t) is the signal and n(t) is AWGN

The input signal and the locally generated φ_1 (t) are available at the product modulator The output of the correletor (Integrate and dump) is used for comparison with a threshold level If the correlator output is greater then threshold decide for Binary 1 otherwise a Binary 0 [22]

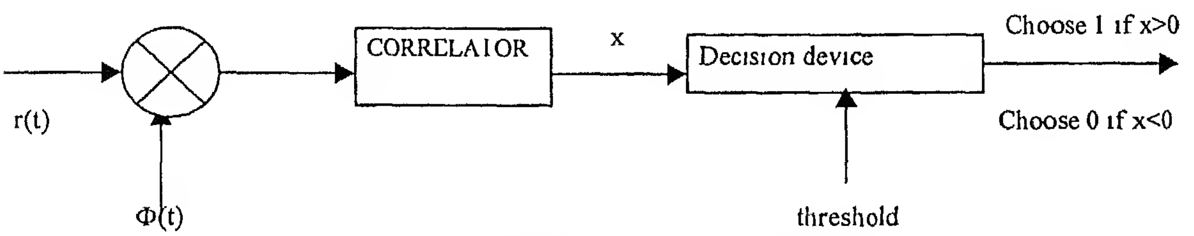


Fig 2 7 COHERENT BPSK RECEIVER

Implementation

1 The 'modulated o/p global variable is the input signal to the receiver The For Loop is used to sequentially give the input pertaining to one bit at a time for further processing

2 The locally generated reference signal is polymorphically multiplied to this input waveform

3 The resultant is given to the integration function

4 The value after the correlator is sampled at T and is compared to a threshold value 0 If greater than 0 the case structure gives a 1 If less than 0 than the case structure gives a 0 as output

5 The O/P bits are the same as the quantised binary input to the transmitter

6 The output bits are given to the "rec O/P' global variable for further processing to follow in the decoder

2 3 2 QPSK RECEIVER

The QPSK signal as defined in section 2 2 is given by the equivalent form in the interval $0 \leq t \leq T$ by

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1)\frac{\pi}{4}\right]\cos(2\pi fct) - \sqrt{\frac{2E}{T}} \sin\left[(2i-1)\frac{\pi}{4}\right]\sin(2\pi fct)$$

and the two basis functions

$$\varphi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi fct) \qquad 0 \leq t \leq T$$

$$\varphi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi fct) \qquad 0 \leq t \leq T$$

The incoming bit stream at the transmitter was partitioned into the Inphase and Quardrature channels The QPSK receiver consists of two correlator channels with the locally generated φ_1 (t) and φ_2 (t) basis functions The output of the correlator of each channel is compared to thresholds and the o/p bits are decided based on this comparison These output bits are then multiplexed to give the original input binary sequence[13]

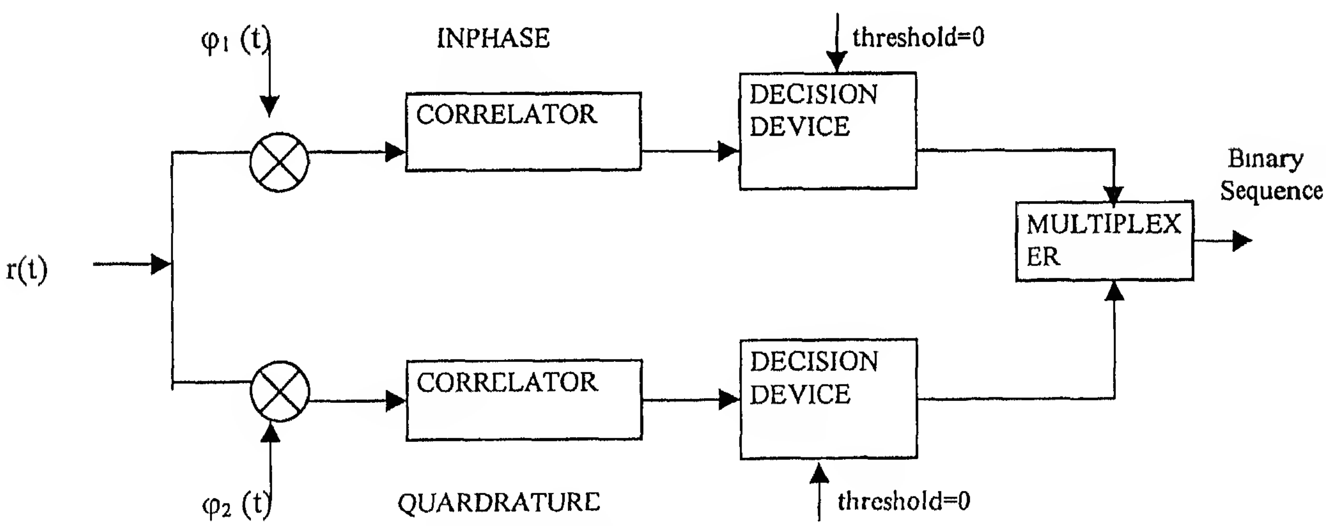


Fig 2 8 QPSK REC

Implementation

1 Convert QPSK signal to a 2 dimensional array form and give to For Loop to process one T sec waveform at one time

2 Generate φ_1 (t) and φ_2 (t) for the Inphase and Quardrature channels

3 Polymorphically multiply the received signal with the φ_1 (t) and φ_2 (t)

4 Integrate the resultant one dimensional array in both channels

5 Index the output of the integrator at T and discard all other values

6 Give output of the For loop to indicators labeled inphase and quardrature

7 Give the indicator O/P through a local variable to a For Loop having a case structure which gives a Binary 1 if input is greater than threshold otherwise 0 The threshold is set to zero

8 The binary output is stored in indicators labeled Even and Odd

9 Use the interleave function using Local variables Even and Odd multiplex to get the desired output This is the detected binary sequence Give this output to chart for plotting

2 3 3 64 QAM RECIEVER

The transmitted waveform is given by [19]

$$S_1(t) = Aa_1\cos(2\pi f_c t) + Ab_1\sin(2\pi f_c t) \qquad 0 \le t \le T$$

The amplitude 7d, 5d, 3d, d, d 3d 5d and -7d of the I and Q signals are assigned 3 bit gray codes 011 010 000, 101 100 110 and 111 respectively The three I and Q bits are denoted by i_1 i_2 i_3 q_1 q_2 q_3 respectively These interleaved bits are demodulated using the decision boundaries The recovery of the bits is based on

$I\ Q \ge 0$	$i_1\ q_1 = 0$
$I\ Q < 0$	$i_1\ q_1 = 1$
$I\ Q \ge 4d$	$i_2\ q_2 = 1$
$4d \le I\ Q < 4d$	$i_2,\ q_2 = 0$
$4d > I\ Q$	$i_2\ q_2 = 1$
$I\ Q \ge 6d$	$i_3\ q_3 = 1$
$2d \le I\ Q < 6d$	$i_3\ q_3 = 0$
$2d \le I,\ Q < 2d$	$i_3\ q_3 = 1$
$6d \le I,\ Q < 2d$	$i_3\ q_3 = 0$
$6d > I\ Q$	$i_3\ q_3 = 1$

Implementation (program # 5)

1 The 64 QAM transmitted signal is given to both the I and Q channels through the For Loop

2 φ_1 (t) and φ_2 (t) are generated using QAM carr vi

3 The input signal is multiplied by the basis function and given to a Butterworth low pass filter of order 2 and sampling frequency set to 100000 (Adjustable based on carrier frequency)

4 The output is indexed at T/2 and stored in indicators labeled RE and IM

5 In sequence Structure 2 this output of the LPF is given to multilevel case structure which will determine the position of the message symbol in the gray coded constellation

6 The i_1 i_2 i_3 q_1 q_2 q_3 bits are determined using earlier defined decision rules

7 The I and Q bits are interleaved as i_1 q_1 i_2 ,q_2 i_3 q_3 This was the 64 QAM symbol transmitted

8 Output of the receiver is given to the "rec o/p " global variable and also to chart for plotting

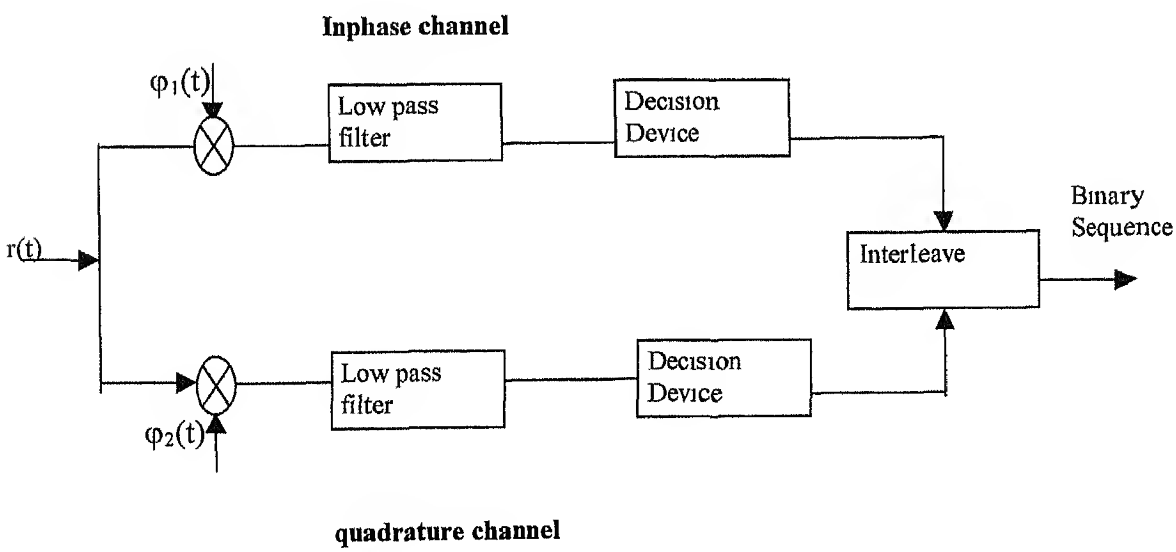


Fig 2 9 64 QAM REC

2 3 4 NON COHERENT FSK

The Quadrature realization of NCFSK has been implemented using correlators in the Inphase and Quadrature channels The correlator o/p are squared added and compared and based on this comparison a binary 0 or 1 output decision is taken[22]

Implementation

1 Generate The basis function $\sqrt{2/T}\cos\omega_1 t$ $\sqrt{2/T}\sin\omega_1 t$ $\sqrt{2/T}\cos\omega_2 t$ and $\sqrt{2/T}\sin\omega_2 t$

2 Multiply the input signal with the above basis functions that is I and Q channels of ω_1 and ω_2

3 Give individual o/p to integrator Index integrator output at T

4 Square the above obtained values and add polymorphically

5 Compare the two added values If the Inphase channel value is greater than the Quadrature channel value than the case structure gives a binary 0 as o/p otherwise a binary 1

NOTE The envelope detector realization of NCFSK is used in practical systems This detector has been implemented in LabVIEW and is placed as an option in the select demodulator vi

2 3 5 MINIMUM SHIFT KEYING

The MSK signal is defined as

$$s(t) = a_I(t)\cos\left(\frac{\pi t}{2T}\right)\cos 2\pi fct + a_Q(t)\sin\left(\frac{\pi t}{2T}\right)\sin 2\pi fct$$

The MSK received signal is multiplied by the respective inphase and quadrature carriers X(t) and Y(t) followed by the integrate and dump circuits The integration interval is of 2T seconds

$$s(t) = a_I(t)\cos\left(\frac{\pi t}{2T}\right)\cos 2\pi fct + a_Q(t)\sin\left(\frac{\pi t}{2T}\right)\sin 2\pi fct$$

$$X(t) = \cos\left(\frac{\pi t}{2T}\right)\cos 2\pi fct$$

Low frequency component of s(t) x X(t) equals

$$a_I(t)\ (1+\cos\frac{\pi t}{T})/4$$

The polarity of the sampler output determines the value of a_I (t) a_Q (t) is determined in a similar manner [24]

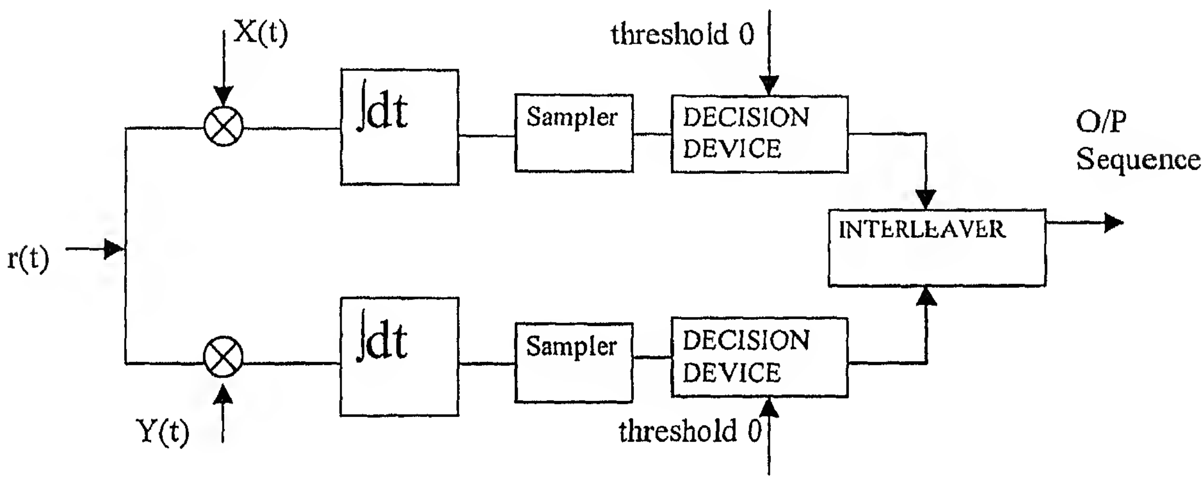


Fig 2 10 MSK REC

<u>Implementation (program # 6)</u>

1 Generate orthonormal basis function using MSK 1 Carr vi and MSK 0 Carr vi

2 Add and subtract the two signals generated to get the basis functions X(t) and Y(t)

3 Set counters l and k to initial values 0 and 100

4 Convert the incoming MSK signal from two dimensional to a single dimension using Reshape Array function

5 Take input from local variable REC waveform over 2T i e 0 to 2T T to 3T, 2T to 4T and so on Correlate with X(t) and sample at 2T Give sampled values to INPHASE DECISION INDEX

6 After each input of 2T increment L by 100 sample points which signify one T

7 For the Quradrature channel take input over 2T from T to 3T 2T to 4T 3T to 5T and so on Multiply with Y(t) and pass to integrator Sample at 2T and give sampled values to QUADRATURE DECISION

8 Increment k by 100 sample points which signify one T

9 Give sampled values to case structure If INPHASE and QUARDRATURE indicator decisions are >0 then a binary 0 is detected If both are < 0 then also a binary 0 is detected For any other combination, a Binary 1 is detected

10 This processing is done for the complete MSK signal produced by an array of 100 bits at the transmitter (considering an input sequence of 100 bits)

11 The detected bits are given to chart for plotting and to the global rec o/p for further processing

NOTE The STOP button in all the modular VI s is to return to the Main Menu

2 4 LINE DECODERS

The next stage after demodulation is to reconvert the binary sequence to the original sequence by using appropriate decoder operation A menu driven option for the selection of the decoder prompts the user to select the decoder An error message is displayed if the wrong decoder selection is made Decoders do the inverse operation of what the coders do The following decoders have been developed

1 Biphase decoder
2 Alternate Mark Inversion decoder
3 3Binary 4 Binary decoder
4 4 Binary 3 Ternary decoder
5 Bipolar Six Zero Substitution decoder

The detailed description of each decoder is present in the VI information of individual programs However one decoder implementation in labVIEW is explained below

2 4 1 AMI DECODER

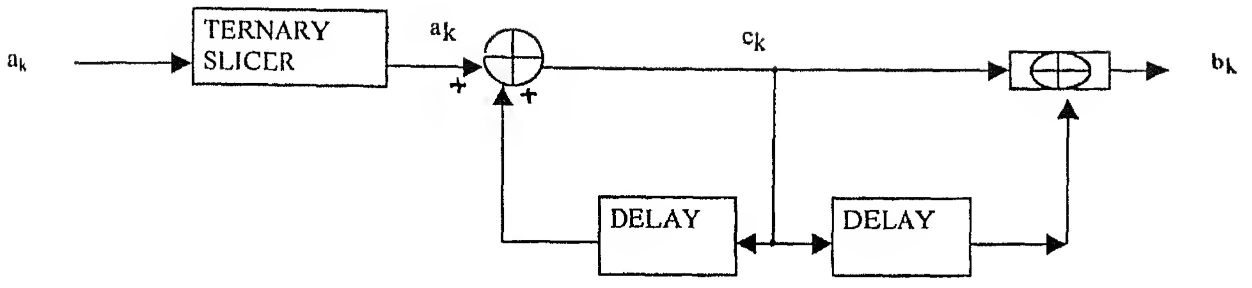


Fig 2 11 AMI DECODER

$$c_k = a_k + c_{k-1}$$

$$b_k = c_k \oplus c_{k-1}$$

$$b_k = (a_k + c_{k-1}) \oplus \hat{c}_{k-1}$$

With various combinations of a_k and c_{k-1} above, The following truth table is made

a_k	b_k
+	1
0	0
	1

Truth Table 2 7 (AMI decoder)

IMPLIMENTATION (program # 7)

1 Reinitialize all input variable to default values

2 Using sequence structure 1 convert all incoming bits to 0 or 1 using the truth table for AMI decoder

3 We still have to extract the first bit Use while loop in sequence structure 2 (for the condition, Do While I = 0 and input bit is not equal to zero)

4 If the input bit is = 1 then decode as 1 else decode as 0

5 The previously decoded bits and this first bit are combined in the build array function to form the decoded AMI bit sequence

6 Give to chart for plotting and to global variable "decoded o/p'

2 4 2 The next stage after decoding is the **reconversion** of the decoded bits to discrete voltage levels and to recover the input waveform, which was quantised modulated and transmitted The various options available for this operation are

1 16 level linear decoder
2 128 level linear decoder
3 A Law decoder
4 μ Law decoder

The algorithm for A Law and μ Law have been described below along with a detailed implementation of 128 level linear decoder

A LAW ALGORITHM

Discrete voltage level = (2Q+1) S=0

$= 2^S(Q+16\ 5)$ S=1 2 7

where S is the **segment** and Q is the **quantisation bracket** Each 8 bit is mapped back to the corresponding voltage level S ranges from 0 to 7 and Q ranges from 0 to 15 (Refer to Table 2 8)

μ LAW ALGORITHM

Discrete voltage level = (2Q+33) (2^S) 33

where S is the segment and Q is the Quantisation bracket Here too each 8 bit is mapped back to the corresponding voltage level (Refer to Table 2 7)

__128 Level Linear Recovery Implementation__ **(Program # 8)**

1 The decoder o/p global output is converted to Boolean using For Loop and Case statement

2 Convert one dimensional array of Boolean into a two dimensional array (100*7) where seven bits signify one voltage level 100 sample points have to be extracted

3 Divide the voltage range by 128 and multiply with array (2^0 2^1 2^2 2^3 2^4 2^5 2^6) which is generated by using the For Loop and scale by power of 2 function Call this the **multiplied array**

4 If first bit is 0 it signifies +ve voltage Use case statement for distinguishing between +ve and ve voltage +ve voltage being processed in case true and –ve voltage in case false

5 Using shift registers multiply input bits with multiplied array components polymorphically and add the one dimensional array components

6 The resultant of the summation gives the voltage level pertaining to the 7 inputs bits

7 Repeat process for all 100 one dimensional arrays of 7 bits and give the output voltage array to chart using the first bit for polarity information

8 The input signal which was initially selected is replicated on chart with the quantisation error (The output signal is still digital It has been interpolated and plotted on the graph)

Note For 16 level decoder same procedure is followed using 4 bits at one time

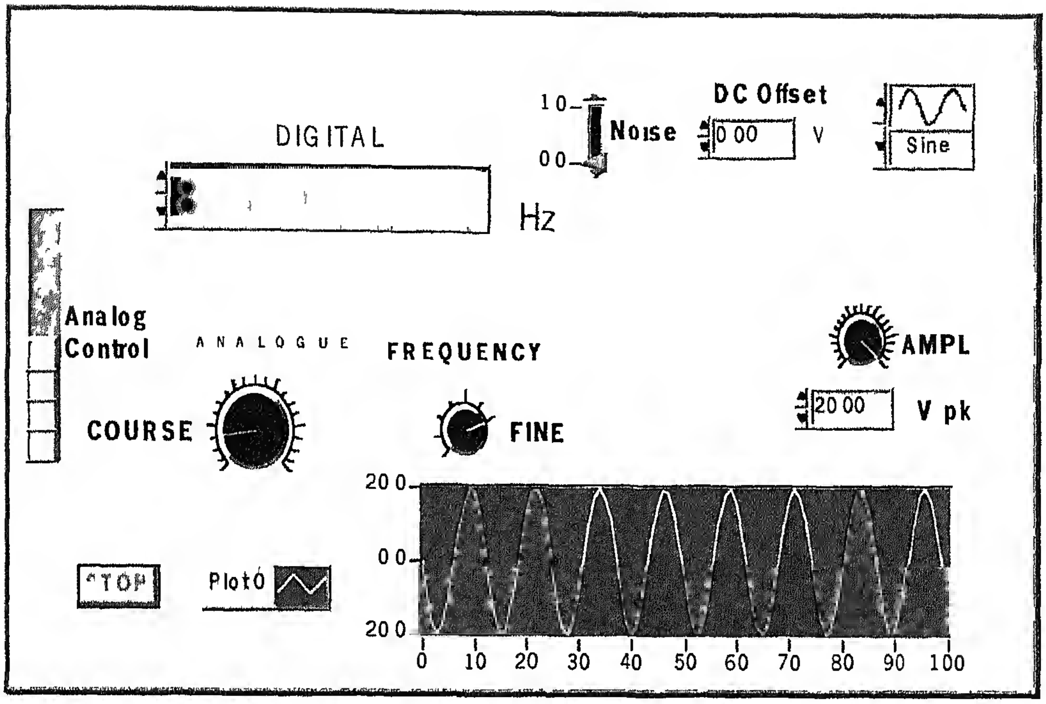


Figure 2 1 Function Generator

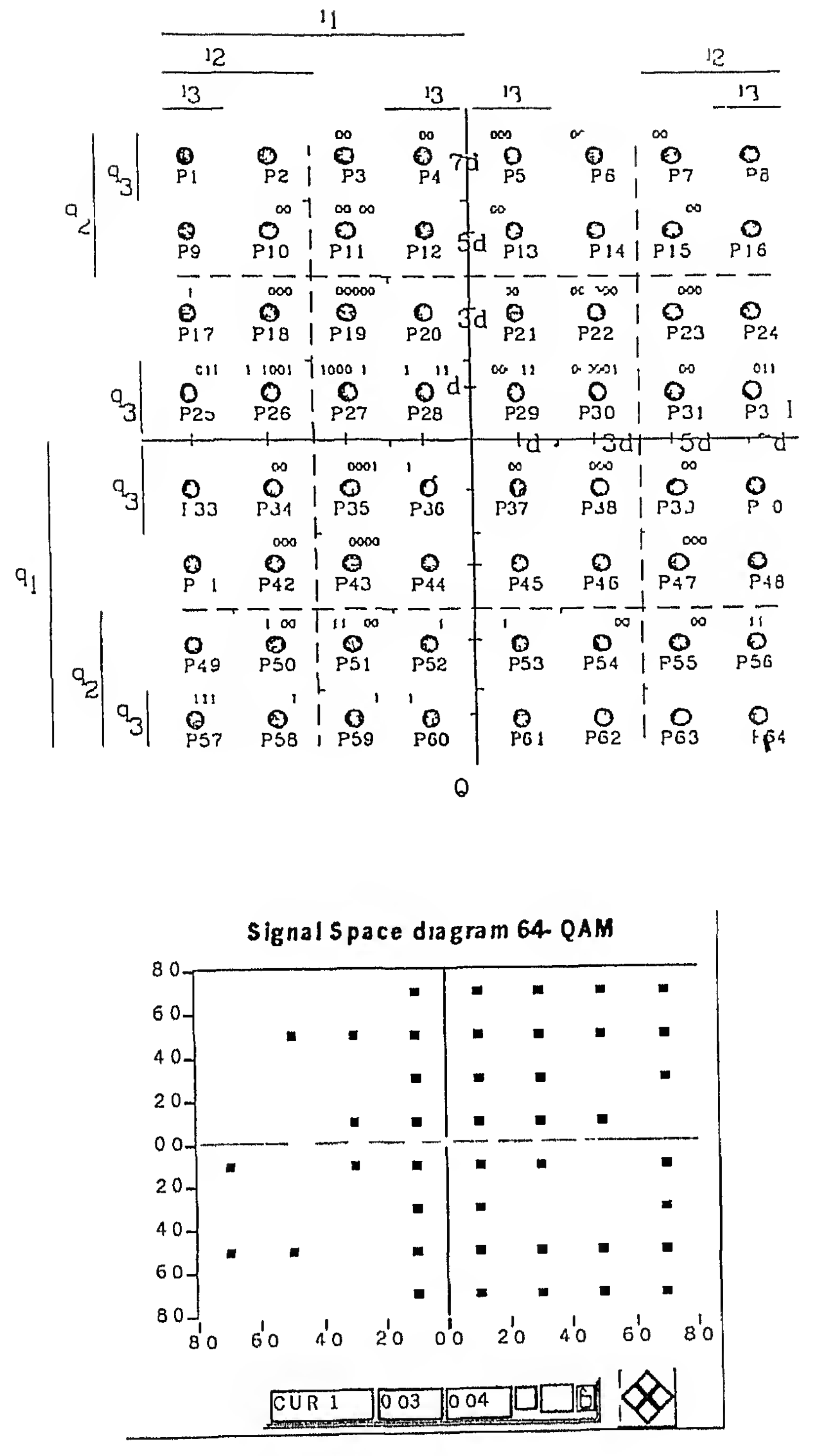


Figure 2 6 64 QAM constellation diagram

Binary No	2's Compliment	Sign Mag		
0000	0	+0	0	0 625
0001	1	+1	0 625	1 25
0010	2	+2	1 25	1 875
0011	3	+3	1 875	2 5
0100	4	+4	2 5	3 125
0101	5	+5	3 125	3 75
0110	6	+6	3 75	4 375
0111	7 (MAX)	+7	4 375	5 V
1000	8(MIN)	0	5	4 375
1001	7	1	4 375	3 75
1010	6	2	3 75	3 125
1011	5	3	3 125	2 5
1100	4	4	-2 5	1 87
1101	3	5	-1 87	1 25
1110	2	6	-1 25	625
1111	1	7	625	0V

Table 2 1 16 level quantisation table

	Source word	Code word	
1	000	0101	0101
2	001	1001	1001
3	010	0001	1110
4	011	0010	1101
5	100	1000	0111
6	101	0100	1011
7	110	0110	0110
8	111	1010	1010

Table 2 3 3B4B encoding table

Binary Input Block	Ternary Output Block		Block Digital Sum
	Mode A	Mode B	
0000	+0	+0	0
0001	-+0	-+0	0
0010	0-+	0 +	0
0011	+-0	+ 0	0
0100	++0	0	±2
0101	0++	0	±2
0110	+0+	0	±2
0111	+++	-	±3
1000	++	+	±1
1001	-++	+	±1
1010	+-+	-+	±1
1011	+00	00	±1
1100	0+0	0 0	±1
1101	00+	00	±1
1110	0+-	0+	0
1111	-0+	0+	0

Table 2 4 4 Binary 3 Ternary encoding table

	Segment Code S								Quantization Code Q	
	000	001	010	011	100	101	110	111		
Quantization endpoints	0	31	95	223	479	991	2015	4063	0000	0
	1	3	103	239	511	1055	2143	4319	0001	1
	3	39	111	255	543	1119	2271	4575	0010	2
	5	43	119	271	575	1183	2399	4831	0011	3
	7	47	127	287	607	1247	2527	5087	0100	4
	9	51	135	303	639	1311	2655	5343	0101	5
	11	55	143	319	671	1375	2783	5[illegible]99	0110	6
	13	59	151	335	703	1439	2911	5855	0111	7
	15	63	159	351	735	1503	3039	6111	1000	8
	17	67	167	367	767	1567	3167	6367	1001	
	19	71	175	383	799	1631	32[illegible]5	6623	1010	10
	21	75	183	99	831	1695	3423	6879	1011	11
	23	79	191	415	863	1759	3551	7135	1100	12
	25	83	199	431	895	1823	3679	7391	1101	13
	27	87	207	447	927	1887	3807	7647	1110	14
	29	91	215	463	959	1[illegible]51	39[illegible]5	7903	1111	15
	31	95	223	479	991	2015	4063	8159		

Table 2 2 μ Law encoding table

	Segment Code								Quantization Code	
	000	001	010	011	100	101	110	111		
Quantization endpoints	0	32	64	128	256	512	1024	2048	0000	0
	2	34	68	136	272	544	1088	2176	0001	1
	4	36	72	144	288	576	1152	2304	0010	2
	6	38	76	152	304	608	1216	2432	0011	3
	8	40	80	160	320	640	1280	2560	0100	4
	10	42	84	168	336	672	1344	2688	0101	5
	12	44	88	176	352	704	1408	2816	0110	6
	14	46	92	184	368	736	1472	2944	0111	7
	16	48	96	192	384	768	1536	3072	1000	8
	1[illegible]	50	100	200	400	800	1600	3200	1001	9
	[illegible]0	52	104	208	416	832	1664	3328	1010	10
	2[illegible]	54	108	216	432	864	1728	3456	1011	11
	2[illegible]	56	112	224	448	896	1792	3584	1100	12
	[illegible]	[illegible]8	116	232	464	928	18[illegible]6	3712	1101	1[illegible]
	[illegible]	[illegible]	120	240	480	[illegible]60	1920	3840	1110	1[illegible]
	[illegible]	[illegible]2	12[illegible]	248	496	992	19[illegible]4	3[illegible]68	1111	15
	32	64	128	256	512	1024	2018	4096		

Table 2 8 A Law encoding table

CHAPTER 3

BASEBAND LINEAR EQUALIZERS

In the previous chapter we saw the LabVIEW implementation of various passband modulation & demodulation schemes In this chapter we simulate ISI using different channel characteristics and pass this ISI effected signal through a Zero Forcing equalizer then a preset type equalizer which uses the LMS algorithm to find the optimal tap gain coefficients and lastly through a hybrid equalizer one that is a combination of a preset and an adaptive equalizer The increase in the noise margins and the effect on detection of the input data through these equalizers have been highlighted through these simulation processes The detailed implementation of the preset equalizer is explained with its LabVIEW block diagram

One of the main factors which contribute to Intersymbol interference is the channel characteristic In Pulse Amplitude Modulation the Intersymbol interference (ISI) results from linear amplitude distortion in the channels that broadens the pulse and causes them to interfere with one another The three channel impulse response characteristics used in simulation are given in Figure 3 1 The first phase of simulation deals with the effect of the channel impulse response and the resultant ISI so produced on the probability of error of detection The eye diagram and the BER plots which constitute the second phase are discussed in chapter 4

3 1 Zero Forcing Linear Equalizers

The total number of taps in the tapped delay line equalizer implemented is (2N+1) The impulse response of the Tapped delay line is

$$h(t) = \sum_{k=-N}^{N} w_K \delta(t - kt) \qquad \delta(t) = \text{Dirac delta function}$$

and w_k = weight at k^{th} tap

If this tapped delay filter is cascaded with a system with impulse response c(t) then the overall response is [13]

$$p(t) = c(t) * h(t) \quad \text{where * stands for convolution}$$

$$= c(t) * \sum_{k=-N}^{N} w_k \delta(t - kT)$$

$$= \sum_{k=-N}^{N} w_k c(t - kT)$$

at t = nT sampling instants

$$p(nT) = \sum_{k=-N}^{N} w_k c((n-k)T)$$

For minimizing **ISI**

$$P(nT) = \begin{cases} 1 & n=0 \\ 0 & n\neq 0 \end{cases} \qquad n=\pm 1\ 2 \quad \pm N$$

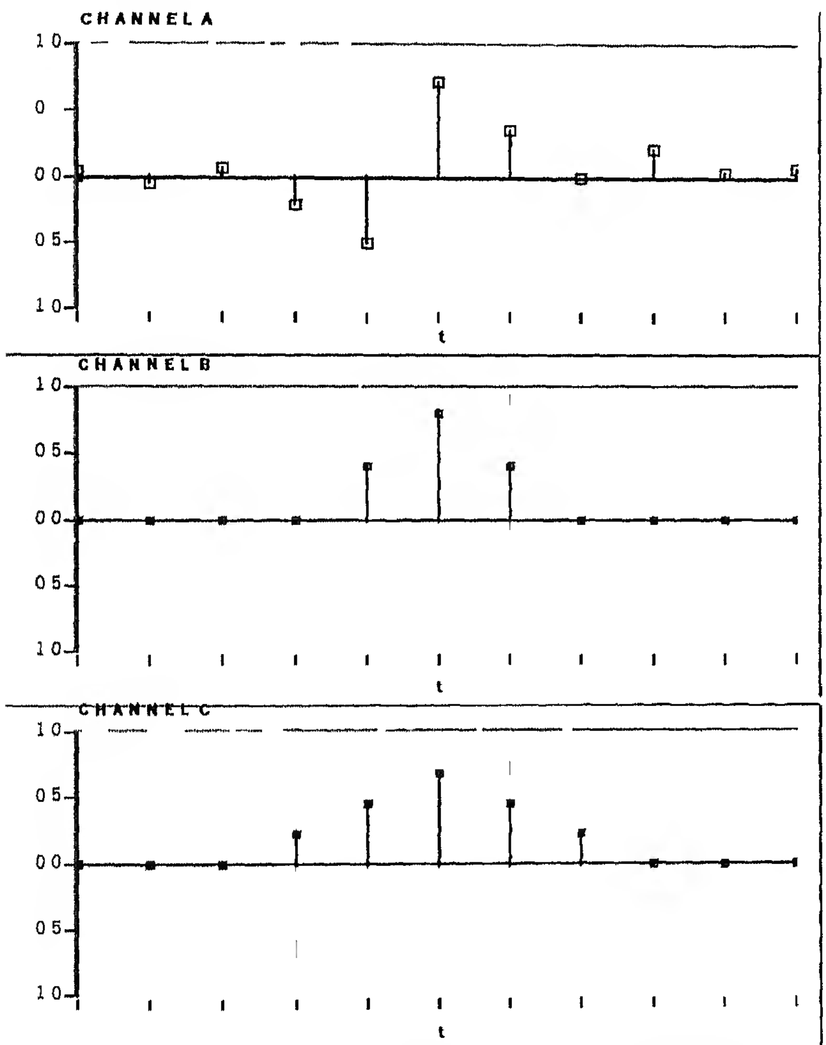


Figure 3 1 Channel Impulse Response

If $c_n = c(nT)$ then the function of the Tapped delay line can be described as

$$\begin{bmatrix} C_0 & - & C & - & C_{-2n} \\ - & - & - & - & - \\ C_1 & - & C_0 & - & C \\ - & - & - & - & - \\ C_{2} & - & C_n & - & C_0 \end{bmatrix} \begin{bmatrix} w_- \\ w_{+1} \\ w \\ w_1 \\ w_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(I)}$$

The larger the numbers of taps the lesser is the distortion

<u>Implementation</u>

1 Generate randomly a binary sequence of suitable length and convert to pulses with time duration T

2 Select any discrete channel from the three characteristics in Figure 3 1 and convolve each binary pulse with the channel Call this h(t)

3 Generate response of the matched filter which matches the channel response Call this h(t)

4 Convolve h(t) and h(t) Repeat this for the complete sequence of input bits

5 Considering only 9 taps on the equalizer sample each h(t)*h(t) output at its peak and at 9T 3T 2T T T 2T 3T 9T Form a 2 dimensional array with these discrete samples for all input bits The basic assumption is that the amplitude at other sampling instants is negligible

6 Shift each one dimensional array by one T as we progress down the two dimensional array That is 2^{nd} Row shift by 1 T 3^{rd} row shift by 2 T and so on

7 Add the discrete values in each column and this is the combined linear effect of ISI What has been described in the above steps is that each pulse is treated individually and these individually effected pulses are linearly added to simulate the overall ISI effect

8 The samples C(nT) are then used to solve the simultaneous equations as described in (I) to determine w_4 w_3 w_0 w_3 w_4 Since N=4 so the dimension of the C matrix is 9 x 9

9 On knowing the coefficients of the Tapped delay filter the output samples are determined by

$$P_o[(k+N)T] = \sum_{n=-N}^{N} C_n P_r[(k-N)T]$$

where P_o (t) is the output sample and P_r(t) is the input to the tapped delay filter

The assumption here is that the pulse broadened due to the channel will have zero interference at sampling instants 4T before and 4T after the peak

10 Each stage of the above implementation is displayed on graphs generated in the labVIEW program

PLOT 1 is the input binary symbols

PLOT 2 is the Channel and transmitted pulse convolution

PLOT 3 is the Matched filter impulse response

PLOT 4 is the Convolution of h(t) and h(t)

PLOT 5 is the detected bits

11 Finally the number of errors between the detected bits and the transmitted bits are determined

12 The detection errors with all three channels with varying number of input bits are discussed in chapter 4 Provision for adding AWGN samples to the transmitted data symbol signal is also available

3 2 PRESET EQUALISERS

In a preset equalizer the error components are measured by transmitting test pulses (PN sequence of maximal length 5 used in the implementation) through the system and inspecting the output of the transversal filter at the sampling instants The equalizer is adjusted prior to or during breaks in data transmission by these predetermined test pulses The Zero Forcing equalization assumes that the initial distortion is less than 100 percent although higher values could be encountered in practice The second approach the rms

equalization where the tap gains must be adjusted such that the cross correlation between the error voltage ϵ_n and the input samples x_n at delays of N N is zero

The mean square distortion of the output response ε^2 is defined as

$$\varepsilon^2 = \frac{1}{h_0^2} \sum_{n}^{1} h_n^2$$

Constraining h_0 using Lagrange multiplier we get [21]

$$\sum_{i\ \infty}^{\infty} 2h_i x_{n-J} = x_J \quad where\ J = -N \qquad N \tag{I}$$

since

$$h_n = \sum_{j\ -N}^{N} c_j x_{i-j} \qquad and \qquad \sum_{\infty}^{\infty} h\ x_{-j} = x_j$$

$$\sum_{k=-N}^{N} c_k b_k = x_J \quad where\ b_k = \sum_{\infty}^{\infty} x\ x_k \tag{II}$$

The best tap gains in the mean square sense are therefore obtained by solving simultaneously (2N+1) linear equations (II)

Let ϵ_n denote the error at sample n present in the output pulse h(t) as compared to the ideal unity height pulse

$$\epsilon_n = h_n \qquad n \neq 0 \tag{III}$$

$$\epsilon_0 = h_0 \quad 1 \qquad elsewhere$$

Then from (I) and (III) we get

$$\sum_{n=-\infty}^{\infty} \epsilon_n x_{n-J} = 0 \quad where\ J = -N \quad N$$

This proves that the **cross correlation between error signal and input pulse should be zero** at each sample point within the range of the equalizer

Implementation

1 Generate PN sequence with m = 5 and initial state of shift registers as 1 0 0 0 0

2 Generate baseband pulses of duration T with Amplitude +1 for a binary 1 and 1 for 0

3 Select channel impulse response A (see figure 3 1) Convolve input pulse with channel impulse response call this h(t)

4 Generate matched filter response with the channel impulse response Convolve with h(t)

5 Preset equalizer being a 7 tap equalizer (In simulation) the discrete amplitude at 3T 2T T T 2T 3T is stored in array form for each incoming binary pulse Then each row of this 2D array is shifted and addition in each column is done The resultant are the ISI effected samples at instants nT (Each pulse effects 3 preceding and 3 succeeding pulses)

6 The output of the matched filter is given to the tapped delay filter where the weights are calculated by the LMS (least mean square) algorithm[18] The weights determined by the LMS algorithm makes the cross correlation between the error signal and the input pulse go to zero

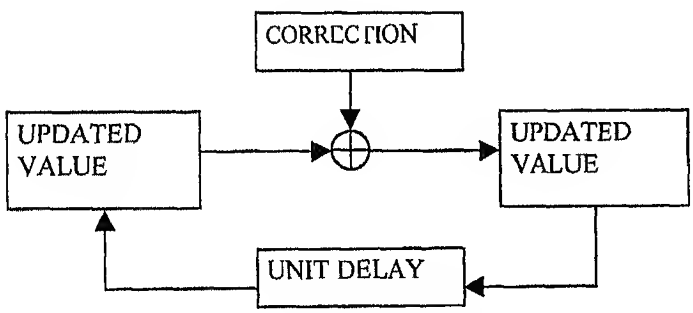


Figure 3 2 LMS Algorithm

$$\underline{W}_{K+1} = \underline{W}_K + 2\mu\varepsilon_K \underline{X}_K$$

where μ is the gain constant that regulates the speed and stability of adaptation $\underline{W}_k$ is the tap gain vector and $\underline{X}_k$ is the input vector to the equalizer filter The LMS algorithm runs iteratively for a selected value of μ till such time the error oscillates around zero The weight vector converges to the optimum vector solution

7 Generate a suitable length binary sequence call this the data sequence Following step 2 through 5 generate the resultant discrete ISI effected samples which will be the input to the Transversal filter

8 With the optimum weights determined by the training sequence carry out operation

$$y_n = \underline{X}_n^T \underline{W}_n$$

where $\underline{X}_n = [x_{n+3} \quad x_{n+1}\ x_n\ x_{n-1}$

$$\underline{W} = [w_3\ w_{-2}\ w\quad w\quad w_3]$$

$$\underline{y}_n = [y_{n+3}\quad y_n\quad y_{n-3}]$$

9 The decision is based on the y_n The threshold is set to zero The detected bits are compared to the input data bits for errors Gaussian noise can also be added to the data bits

10 Output graphs at each stage as mentioned earlier can be viewed at each stage in the labVIEW program

3 3 ADAPTIVE EQUALIZATION

In adaptive equalization the error voltage ϵ_n are continuously estimated during the course of normal data transmission and corrections to the equalizer coefficients are effected as required [21] Such closed loop system has an inherent accuracy advantage over the preset type of equalizer In random data transmission the transmitted sequence is not available but the estimate at the output of the threshold detector is used as if it were the correct transmitted sequence $\{a_n\}$ The error voltage is given by

$$\epsilon_n = a_n \quad y_n$$

where y_n is the equalizer output at time nT and a_n is the estimate after the threshold detector

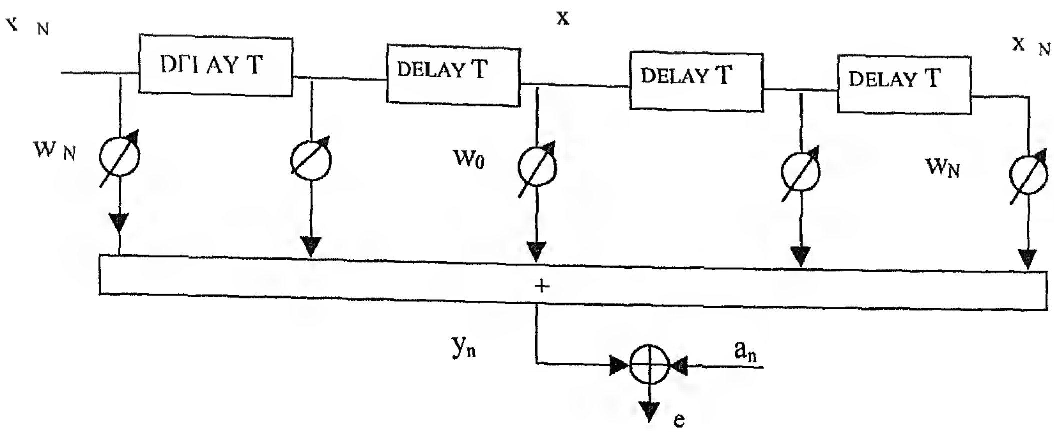


Figure 3 3 Elements of an Adaptive Filter

Implementation of Decision Directed Equalizer

1 Step 1 through 6 in the preset equalizer implementation are same for the decision directed mode equalizer Once the optimal tap coefficients are determined generate a suitable length of randomly generated binary numbers using the iterative For Loop and the random no generating function

2 Each pulse is transmitted over the channel and the resultant input to the equalizer is

C(t)*h(t)*h(t) Where * stands for convolution

$\underline{Y}_i = \underline{X}_n^T \underline{W}_n$ where $\underline{X}_n$ are the input to the equalizer $\underline{W}_n$ is the tap gain coefficient vector and $\underline{Y}_n$ is the response of the equalizer

3 The threshold detector (case statement) gives output a_n 1 or 1 The difference a_n yn is determined and the tap weights are readjusted using this value of ϵn The modified weights would result in a lower error ϵn

4 With each detection of the transmitted pulse the correction to the weights is done

5 The impulse response of the channel was varied at a slow pace by a magnitude of 10^{3} at each discrete point of it s response with each transmitted bit In one case the channel was tracked and in the other left untracked With varying noise variance it was seen that the data transmitted through the non tracked channel gave 5 times more errors than the tracked channel where the optimal weights were changing dynamically

This program includes plots of this varying channel and the error indicators of the tracked as well as the untracked channel The basic stage wise plots as before are present as well

The Preset channel vi (program # 9)

1 Sequence structure 0 is used for initializing all input variables to default values

2 Sequence 1 generates the PN sequence and converts the generated symbols to polar 1 and –1 using baseband vi AWGN Noise samples are generated which are added to the data samples

3 Sequence 2 convolves the input pulse and the channel response and further convolves the matched filter response with the resultant of the above convolution

4 Sequence 3 shifts the output at the matched filter by the corresponding T and then transposes this 2 dimensional array and adds in a row wise fashion It generates the input to the equalizer filter

5 Sequence 4 first runs the LMS algorithm and determines the optimal weights The input vector to the equalizer filter and these weights are added and a detection decision is taken whether a binary 1 was transmitted or a binary 0

6 Sequence 5 generates the data bits using the random number generation function The length of the binary sequence is specified by the no of bits control Step 2 through 4 repeat for the data sequence and the output of the matched filter is given to the equalizer

7 Sequence 6 multiplies this input vector with the optimal weights determined earlier

8 The output of the sum array element function is given to a case statement where the binary sequence is reconstructed

9 In case of the adaptive equalizer vi the difference between a_n and y_n is given as the error value to correct the optimal weights The next set of input to the 2N+1 tap equalizer filter use these changed gain coefficients This is a closed loop process where after each detection of a bit the weights are adjusted

The various processing stages of the preset vi are depicted in the **flow chart** placed at the appendix

CHAPTER 4

PARAMETER DETERMINATION AND SIMULATION RESULTS

In the previous two chapters we have discussed the simulation of various passband modulation schemes and equalizers in the baseband case. In this chapter **Section 4.1** we discuss the various parameters to be determined such as the amplitude spectrum, power spectrum of the input signal and the modulated carrier signal, the response of the filters used in the detectors, the Gaussian pdf of the noise samples and the quantisation parameters. The plots of these above parameters obtained through LabVIEW implementation are also discussed. The BER plots of some modulation schemes are discussed in **Section 4.2**. In **Section 4.3** the output signal generated by QPSK, OQPSK and MSK schemes are compared and a comparison of the errors generated by bandlimited OQPSK and QPSK modulation with varying noise variance is discussed. The eye diagram of the equalizer output and the effect of channel characteristics are discussed in **Section 4.4**. **Section 4.5** deals with the regeneration of an externally generated signal, using BPSK modulation scheme, on a CRO.

4.1 PARAMETERS

In the passband modulation demodulation capsule following parameters of the output processes are determined:

i) Amplitude and power spectrum of the input signal

ii) Amplitude and power spectrum of the modulated carrier signal

iii) Quantisation parameters

iv) Filter response

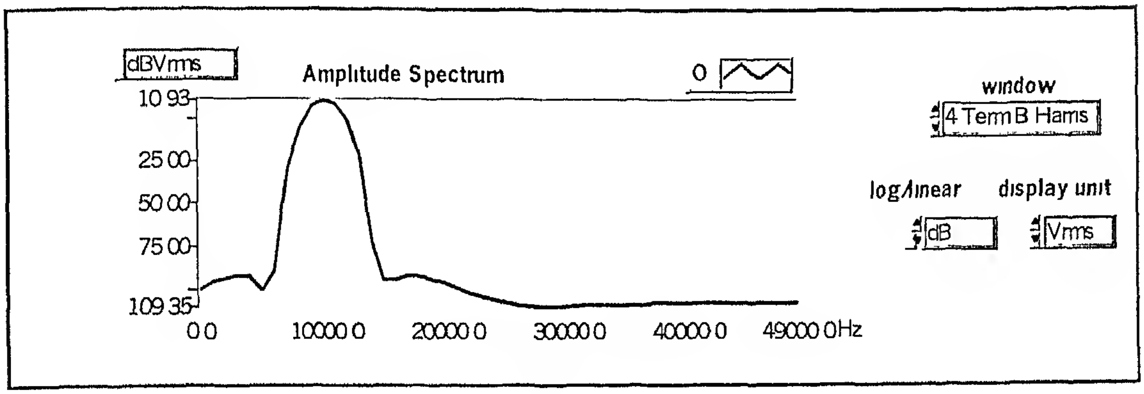
v) PDF of AWGN noise samples

4.1.1 Amplitude and Power Spectrum

The amplitude and power spectrum is computed as

$$\frac{\text{FFT(Singal)}}{(N)}$$

where N is the number of samples points in the signal The program then converts the amplitude to single sided rms magnitude spectra Plot I shows the amplitude spectrum of a 10 kHz signal

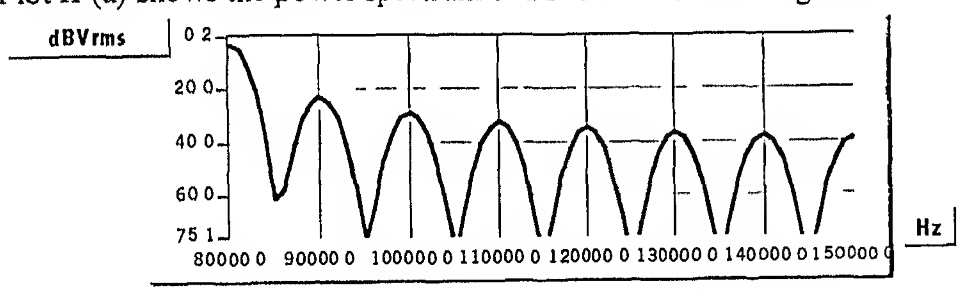


Plot I

The power spectrum is computed as

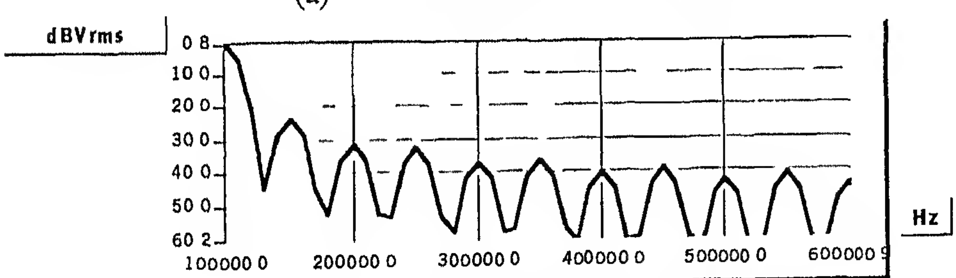
$$\frac{FFT*(Singal)xFFT(signal)}{(N^2)}$$

where N is the number of points in the signal and * denotes the complex conjugate The program then converts the power spectrum into a single sided power spectrum Plot II (a) shows the power spectrum of a MSK transmitted signal at 80000 Hz



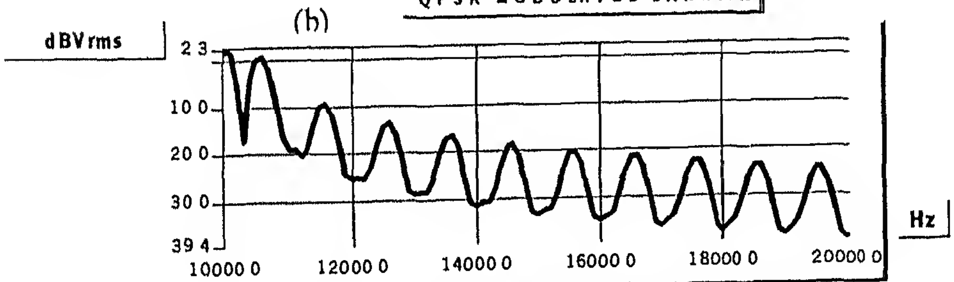
MSK MODULATED CARRIER

(a)



QPSK MODULATED CARRIER

(b)



(c) 16 QAM MODULATED CARRIER

PLOT II

The plot II (b) and (c) above show the power spectrum of a QPSK signal at 100000 Hz and 16 QAM modulated carrier at 10000 Hz

Note Both the power spectrum plots use 1500 samples
In all the plots above the units and the scales (logarithmic / linear) can be selected as per the options available on the front panel

4 1 2 <u>Quantisation parameters</u>

The following parameters of the linear quantizers used are determined in the quantiser parameter vi and are displayed on its front panel using slide indicators and formula nodes

i) a (spacing)

ii) Mean square error ($a^2/12$)

iii) Root mean square error ($a/\sqrt{12}$)

iv) Mean signal power (a^2) $\left[\dfrac{M^2-1}{12}\right]$ where M is the level of quantisation

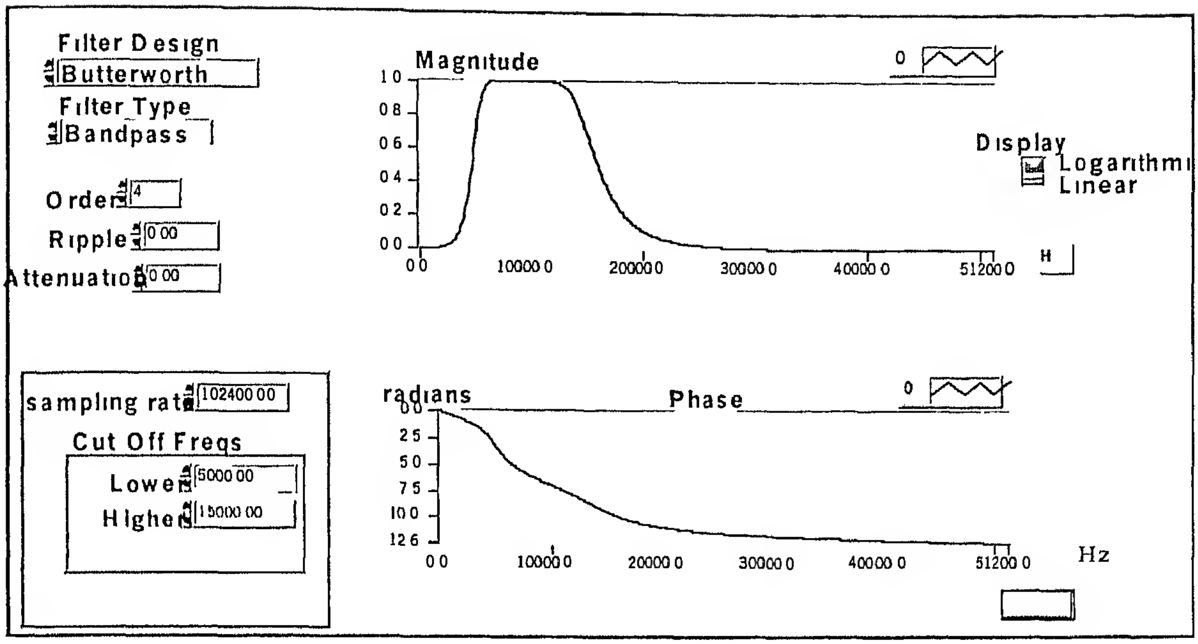
v) Signal to noise ratio (peak/rms) $V/(a/\sqrt{12})$

4 1 3 <u>Filter response</u>

This program graphically shows the transfer function of a butterworth filter used as a band pass or a low pass filter The order sampling rate and cut off frequencies (low and high) have to be specified as inputs parameters to the programs These parameters are set to some default values The functioning of the program is as follows

1 Generate 1024 samples of impulse pattern
2 Pass through desired filter (Low pass or Band pass)
3 FFT of filtered output is converted to polar form
4 The magnitude gives the impulse response plot and the angle gives the phase response plot

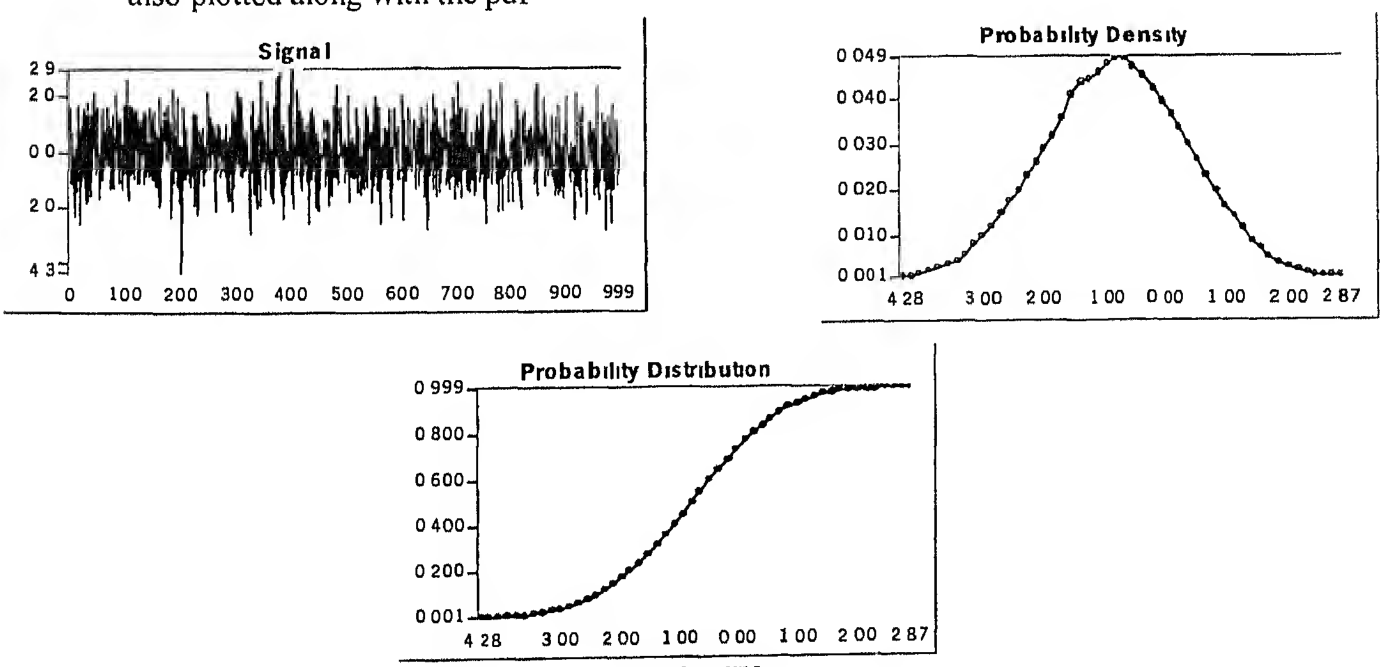
Plot III shows the magnitude and phase spectrum of a bandpass butterworth filter The order and the cut off frequencies are set to some default values



PLOT III

4 1 4 Distribution and Density Function of AWGN samples

This program determines the probability distribution of the AWGN samples using the histogram and integration functions available in LabVIEW The density function is determined by taking the derivative of the distribution function The minimum samples required to generate a bell shaped density function is 1000 Plot IV gives the pdf and the distribution of 1000 samples of AWGN The input samples are also plotted along with the pdf



PLOT IV

4 2 PROBABILITY OF ERROR

For digital communication system the relevant measure of performance is always related to the systems error producing behavior BER is the fractional number of errors in a transmitted sequence The conceptual framework within which this problem can be visualized is as follows The input is defined as an ensemble of identical systems the input to each of which is a distinct joint realization of the signal and noise source The input to the decision device is this set of waveforms and the output is a digital sequence which in general will contain some errors For the kth member of the ensemble the error rate is defined as

$$P_k = \lim_{N\to\infty} n_k(N)/N$$

Not all P_k will be identical since these are realizations of the digital sequence which are not typical The setting within which P is defined is identical to the classical Bernoulli trails with probability P of success and (1 P) of failure

"Monte Carlo is merely a label for the implementation of a sequence of Bernoulli Trials in which we count the number of errors and divide by the number of trails This method requires no assumptions about the input processes or the system

BER for the assumed symbol can be written as

$$P = \int_{\upsilon\in D_0} f_V(\upsilon)d\upsilon$$

where f_v is the pdf of the sampled symbol at sampling epoch and D_o is the region of v corresponding to an error

Define error indicator function as

$$H(\upsilon) = \begin{cases} 1 & \upsilon\in D_0 \\ 0 & \upsilon\notin D_0 \end{cases}$$

$$P = \int_{-\infty}^{\infty} H(\upsilon) f_v(\upsilon)d\upsilon$$

$$P = E[H(v)]$$

The estimator P of the expectation is the sample mean we have

$$P = \frac{1}{N_o}\sum_{i\in I} H(V_i)$$

where N_o is the number of elements and V_i is the sequence of symbol space samples of the decision voltage I_o is the integer set that contains all i

If all symbol have the same probability of occurrence then

$$P = n(N)/N$$

where N are the total symbols & n is total errors

As N→∞ P → P a s

To generate the BER graphs for various modulation schemes the following guidelines and assumptions are used

1 20000 data bits are generated from a random source for processing

2 AWGN samples with varying seed are added to the input waveform generated from the data bits

3 Averaging over 5 times at each input variance value is done to smoothen the BER curve

{Due to limitations of processor larger set of data bits could not be considered}

4 For the correlator receivers No/2 = σ^2 where No/2 is the single sided noise density in watt/Hz [23]

4 2 1 Probability of Error BPSK

$$P_{eBPSK} = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

Practical Implementation

BPSK Error vi (**program # 10**) runs the PSK Verific vi, which calculates the number of bits in error at varying input noise variance

1 Randomly generate input signal, which is quantised by a 4 level linear quantiser producing 400 bits per input signal

2 The input bit stream is given to a BPSK modulator which generates the carriers for the binary 0 and 1

3 AWGN samples at desired (varying) variance is added to the carrier and this carrier plus noise is given to the receiver

4 The correlator realization of the BPSK receiver detects these bits according to the polarity at the output of the threshold detector

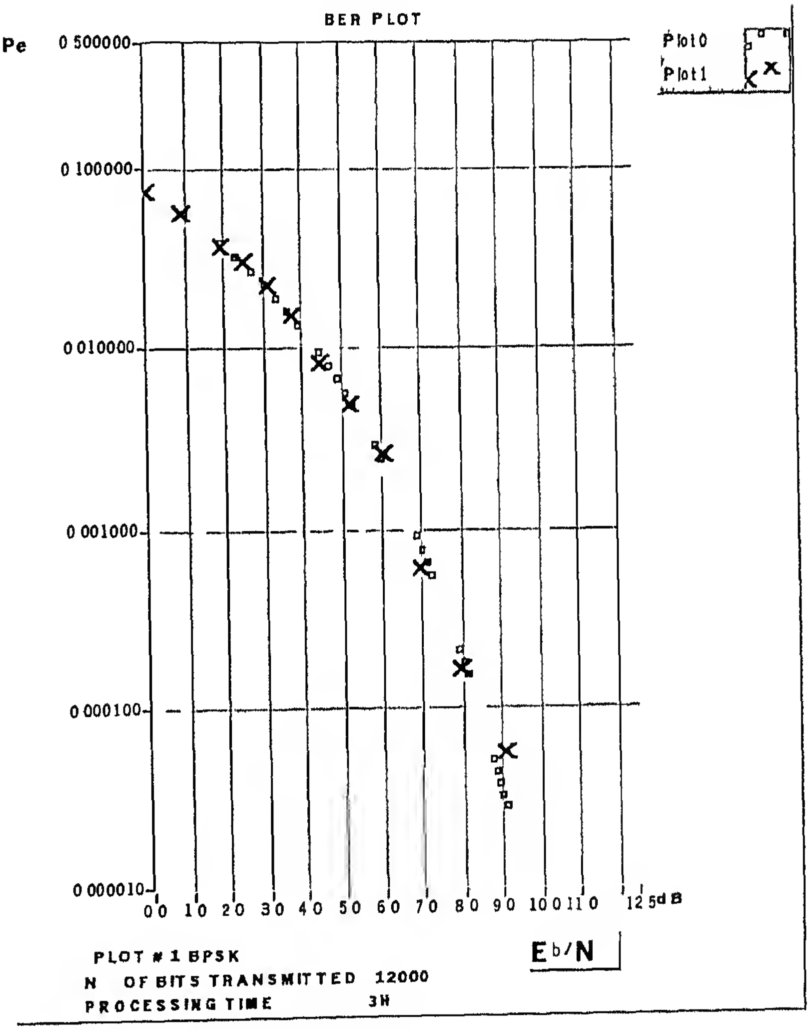
5 The input bit sequence is compared to the detected bit and the number of bits in error are determined for each run The number of bits selected is between 10000 to 20000 At each variance averaging is done over 5 runs

6 The average signal power to average noise power (SNR) is dB is the x axis of the plot with the y axis being the bits in error to total bits The plot is in logarithmic scale

7 The BER plots had to be extrapolated for higher values of SNR due to limitations of simulating a run with larger bit sequence The processing time for a sequence 10000 bit long would take upto 3 to 4 hours

8 The theoretical plot of Pe and Eb/No is plotted (plot#1) along with the practical plot using the expression

$$P_e = \frac{1}{2}\operatorname{erfc}\sqrt{\frac{Eb}{No}}$$



NOTE In plot #1 to plot # 4 plot 0 is the theoretical plot and plot 1 is the simulated plot

4 2 2 Probability of Error QPSK

$$P_{QPSK} = \frac{1}{2} erfc \sqrt{\frac{Eb}{No}}$$

QPSK transmits it twice the bit rate of a coherent binary PSK system for the same channel bandwidth

The plots of BER Vs SNR obtained practically and theoretically are given in Plot#2

The QPSK error vi runs the same way as BPSK error vi with QPSK modulator and receiver

4 2 3 Probability of Error DPSK

$$P_e = \frac{1}{2} \exp\left(\frac{-E}{2No}\right)$$

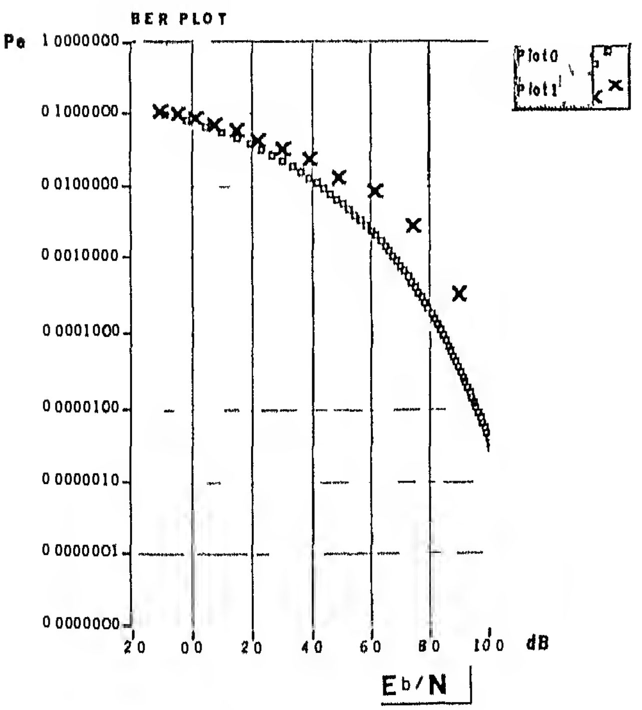
No = power spectral density

$$E = 2E_b$$

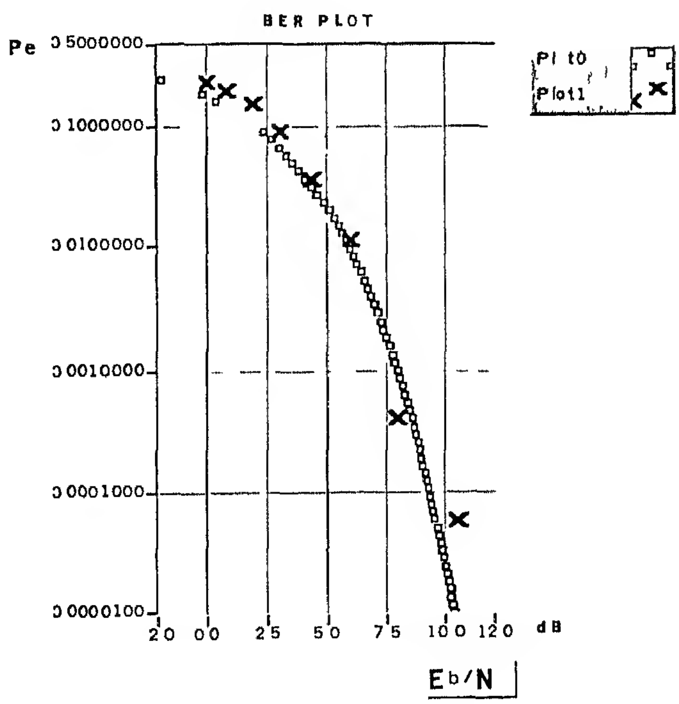
$$P_e = \frac{1}{2} \exp\left(\frac{-Eb}{No}\right)$$

E_b = signal energy per bit

The Simulated and the theoretical plot are given in Plot#3 The practical implementation has been done using DPSK Error vi using DPSK transmitter and receiver with AWGN samples



PLOT # 2 QPSK
N OF BITS TRANSMITTED 12000



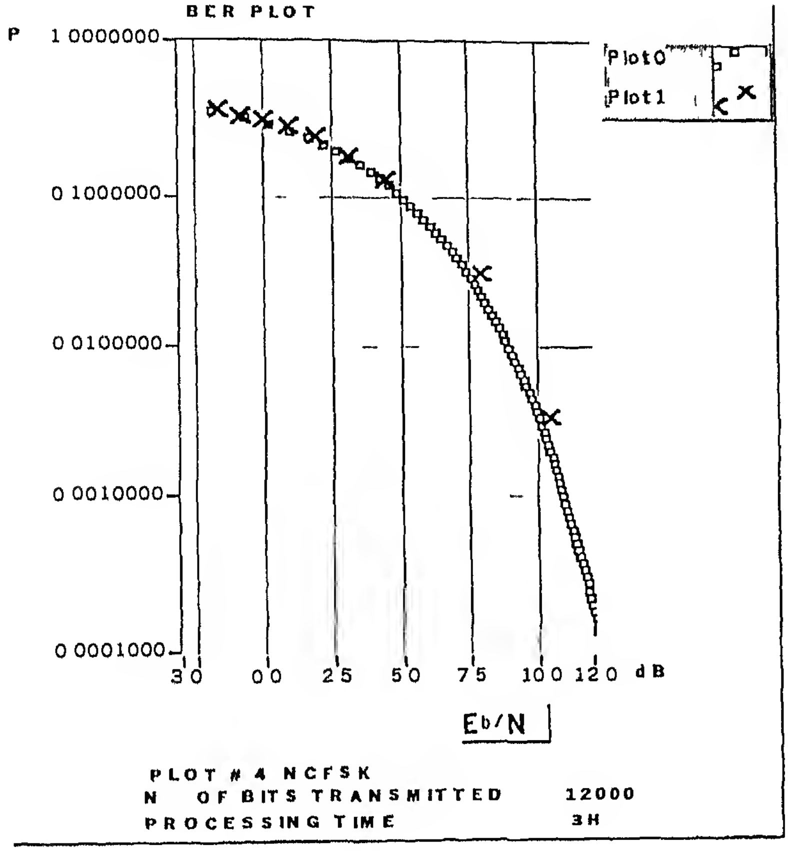
PLOT # 3 DPSK
N OF BITS TRANSMITTED 12000

4 2 4 Probability of Error Non Coherent Frequency Shift Keying

Plot#4 gives the probability of error Vs SNR plot of noncoherent FSK found practically and theoretically

The theoretical plot is found using

$$P_e = \frac{1}{2}\exp(-Eb/2No)$$



4 2 5 Probability of Error 16-QAM

The positions of the bits in the QAM symbols associated with each point in QAM constellation have an effect on the probability of them being in error Of the four bits in 16 QAM, for i_1 and q_1 (MSB), the distance from the demodulation decision boundary of each received phasor in the absence of noise is 3d for 50% of the time and d for the rest i_2 and q_2 (LSB) are always at a distance d 16 QAM may be considered as having class C1 and class C2 subchannel where C1 has lower probability of error than C2 [19]

For **C2 subchannel**

$$P_{2C} = Q\left(\frac{d}{\sqrt{No/2}}\right) = \frac{1}{\sqrt{2\pi}} \int_{\frac{d}{\sqrt{No/2}}}^{\infty} \exp(-x^2/2)dx$$

$E_o = 10d^2$ [Square constellation with gray coding]

For **C1 subchannel**

$$P_{2C} = Q\left\{\sqrt{\frac{Eo}{5No}}\right\} \quad \text{and}$$

$$P_{1C} = \frac{1}{2}\left[Q\left\{\frac{d}{\sqrt{No/2}}\right\} + Q\left\{\frac{3d}{\sqrt{No/2}}\right\}\right]$$

$$= \frac{1}{2}\left[Q\left\{\sqrt{\frac{Eo}{5No}}\right\} + Q\left\{3\sqrt{\frac{Eo}{5No}}\right\}\right]$$

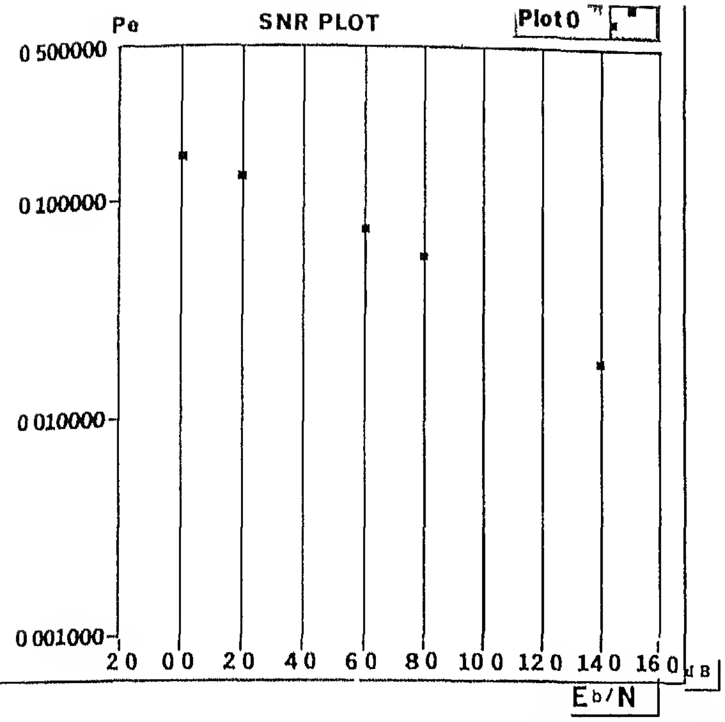
Combining P_{2G} and P_{1G}

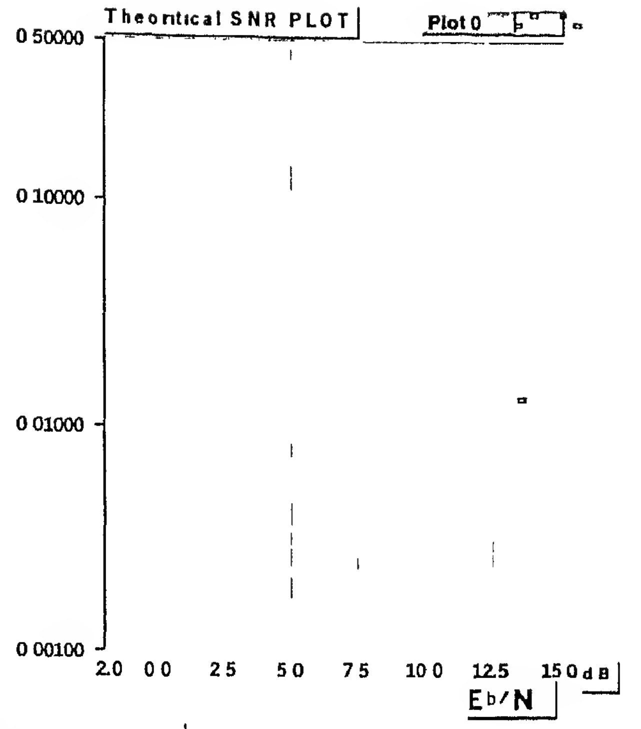
$$P_{av} = (P_{1G} + P_{2G})/2$$

The theoretical plot has been based on the above expressions

Passing bits through the 16 QAM modulator AWGN channel and demodulator chain and checking for errors at varying noise variance we obtain the practical results The seed of the AWGN sample generator has been varied in the generation of the gaussian samples Plot#5 shows the BER for 16 QAM

NOTE The theoretical plot is the symbol error Vs Eo/No plot where Eo is the average energy and No is the spectral density of noise The low pass filter realization of the detector which is used is not the optimum detector realization





PLOT # 5 16 QAM

N OF BITS TRANSMITTED 12000

PROCESSING TIME 3 H

4 3 QPSK, OQPSK AND MSK SIGNAL GENERATION

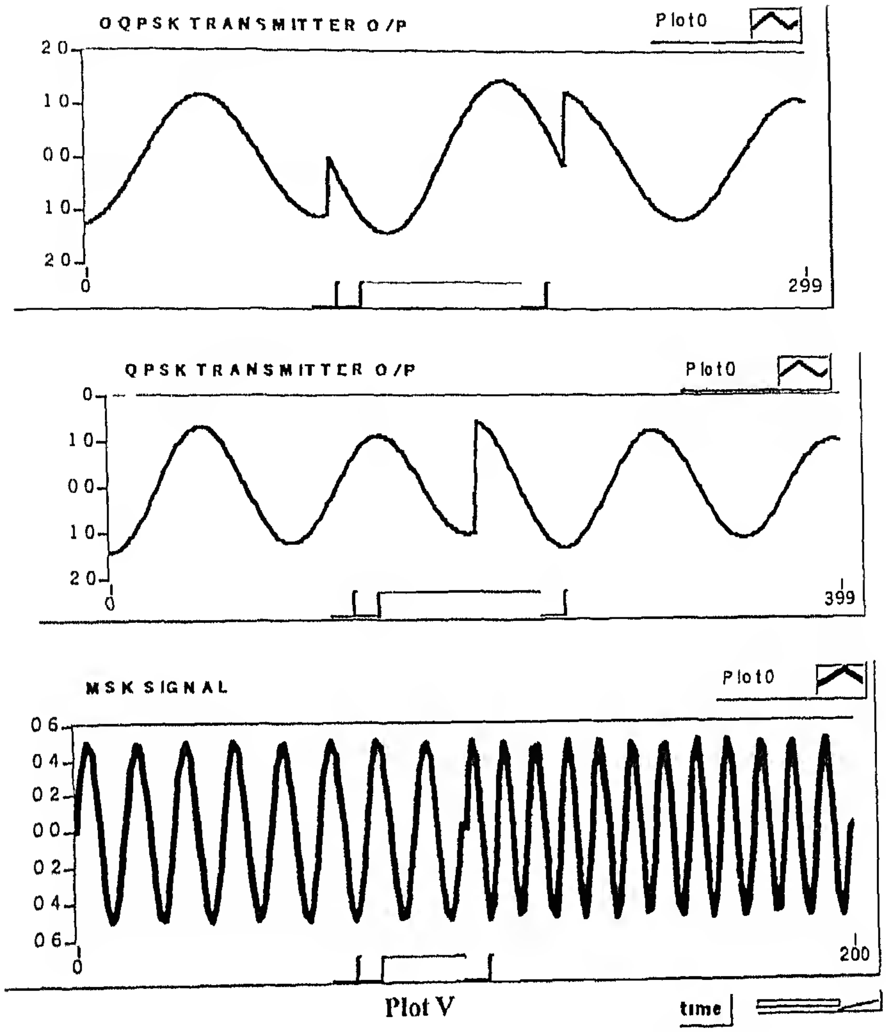
4 3 1 In this section we compare the QPSK signal OQPSK signal and the MSK signal, generated at the respective transmitters

The MSK signal has no transitions of phase and it is continuous in phase

The QPSK signal has a maximum phase transitions of 180° if both Inphase and Quardrature component change phase simultaneously

The OQPSK can have a maximum phase transition upto ±90° due to the offset by T_b in the inphase and quardrature channel

Plot V shows these phase transitions in each of the transmitted signal



Plot V

4 3 2 QPSK, OQPSK Through Bandpass Filter

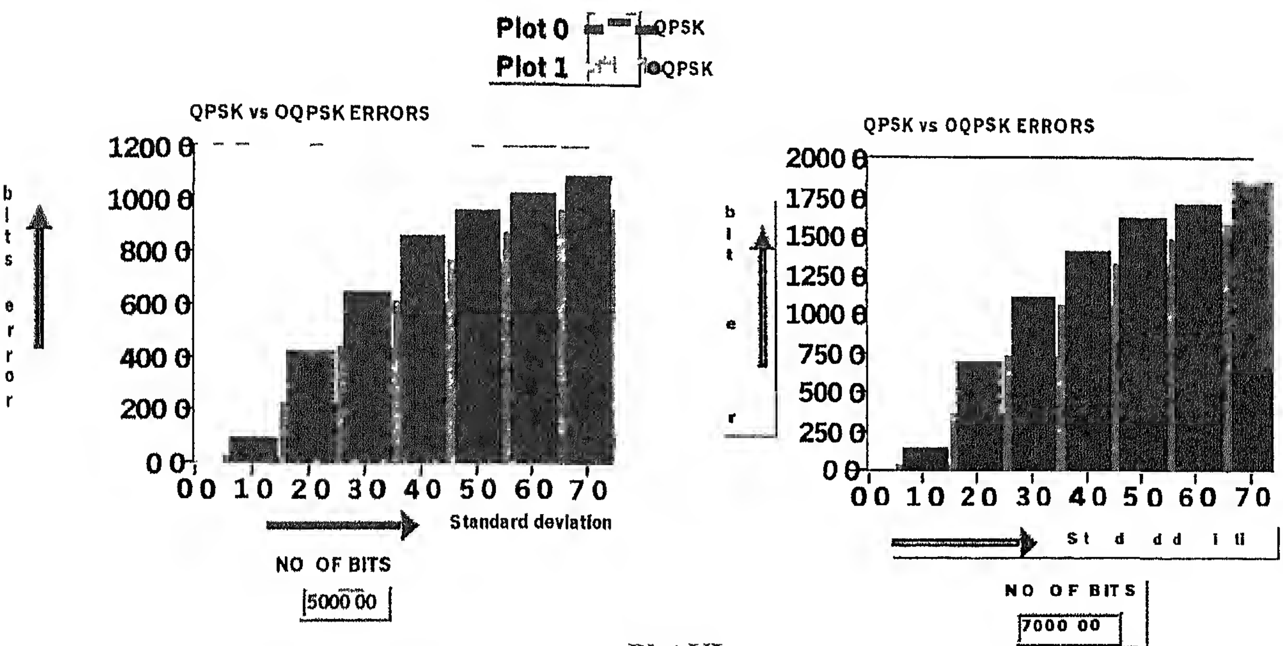
The histogram plot VI below give the comparison of the errors generated in the QPSK modulation scheme and the OQPSK modulation schemes by varying the gaussian noise variance and passing them through a bandpass filter Comparison with a run of 7000 bits and 5000 bits are shown in the plot It is observed that OQPSK modulation scheme generates fewer errors than the QPSK modulation scheme

Implementation

1) A 7000 bit long input binary sequence is randomly generated (f1 and f2 for the carriers is 100000 Hz High cut off frequency of the bandpass filter is 195000 Hz Low pass frequency cut off is 5000 Hz The order of the butterworth filter is 2)
2) This is given to the QPSK transmitter, which generates the QPSK signal This signal is bandpass filtered and given to the detector AWGN noise samples are

added to the transmitted signal At the detector the coherent carrier so generated is also passed through a bandpass filter which has the same high frequency cut off and low frequency cut off as the QPSK signal

3) The detection errors at varying noise variance are recorded in a spreadsheet format file
4) The OQPSK signal generated is passed through a bandpass filter and detected at the detector Here too the coherent carrier so generated is passed through a band pass filter with the same parameters The detection errors at varying noise variance are recorded in a spreadsheet format file
5) The errors at various noise variances are compared for QPSK and OQPSK a histogram plot is generated for a direct comparison
6) It is observed that the OQPSK generates less errors then QPSK



Plot VI

4 4 EQUALIZERS AND EYE DIAGRAM

4 4 1 Channel Impulse Response and Errors Due to ISI

The channel impulse response plays an important role in the generation of ISI In some channels the ISI can be so severe that even in the absence of noise detection errors persist Figure 3 1 gives the three channel impulse response used in the

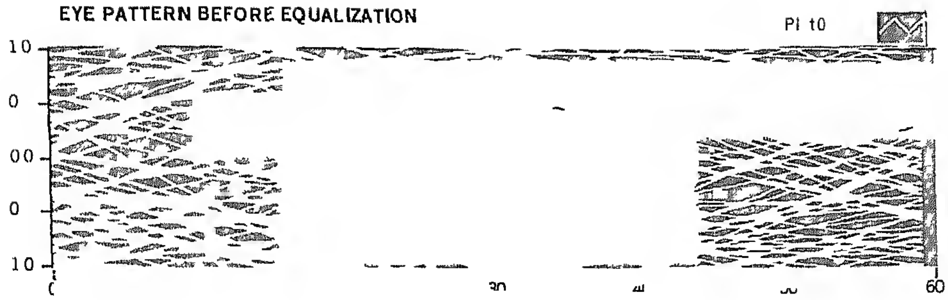
simulation The ISI pulse so constructed (Refer Implementation of Zero forcing Equalizer) is given to the ZF equalizer using all three channel characteristics The errors detected with varying input bit size is recorded in ZFE spreadsheet file Table 2 8 brings out the detection errors for various runs of the simulation for all the three channel impulse response

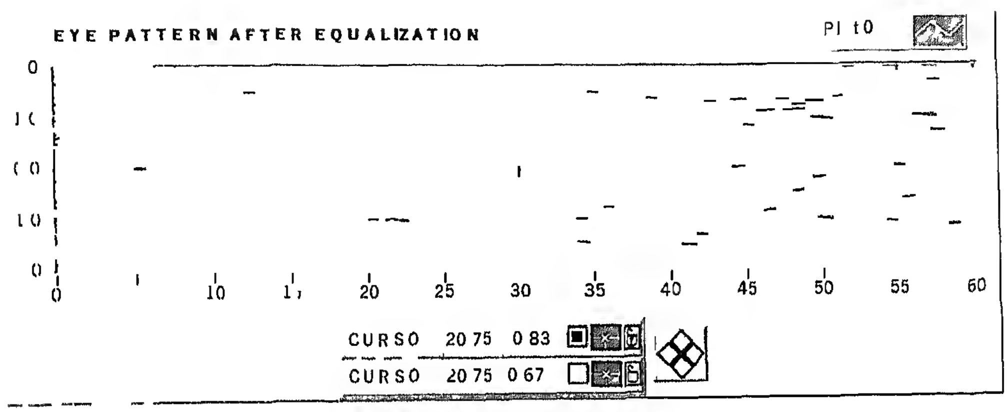
7	22	35	43	94		CHANNEL	B
0	0	0	0	0		CHANNEL	A
6	12	32	42	86		CHANNEL	C
100 bits	200 bits	400 bits	500 bits	1000 bits			

Table 2 9 **Bit errors at various runs**

4 4 2 Eye Diagram Generation

Refer to implementation of adaptive equalizer in Chapter 3 The signal before the transversal filter and the signal after the transversal filter are used to generate the eye patterns before and after equalization The interpolated signal has been fragmented into signal over T sec These T sec signals are superimposed (stacked) one over the other The input signal over 100 T (100 bits generated) when stacked generates the eye diagram before equalization The eye diagram after the equalizer is generated using the output signal from the equalizer (Interpolated) and again superimposing (stacking) T sections of the signal It is observed from the eye diagrams simulated that there is an improvement in the noise margin in the signal after the transversal filter Plot VII shows the eye diagram before and after equalization The data for constructing the eye patterns has been extracted from the Preset equalizer vi





PLOT VII

4 4 4 Probability of Error For PAM With ISI

If a time limited signal is limited to N L ary symbols the ISI can only assume L^N possible values However if the impulse response x(t) is time limited to k symbols before and after t_0 then L^{2k} sequences need to be considered to evaluate the probability of error The probability of error can be determined by averaging over all L^{2k} sequences [21]

$$P_e = \sum_{i=1}^{L^{2k}} P_{ei} P(i)$$

P_{ei} is the probability of error for the i^{th} sequence and P(i) is the probability of occurrence for the i^{th} sequence

Let $\sum_{n} a_n x_{-n} = D(i)$ where $D(i)$ is the distortion over ith sequence

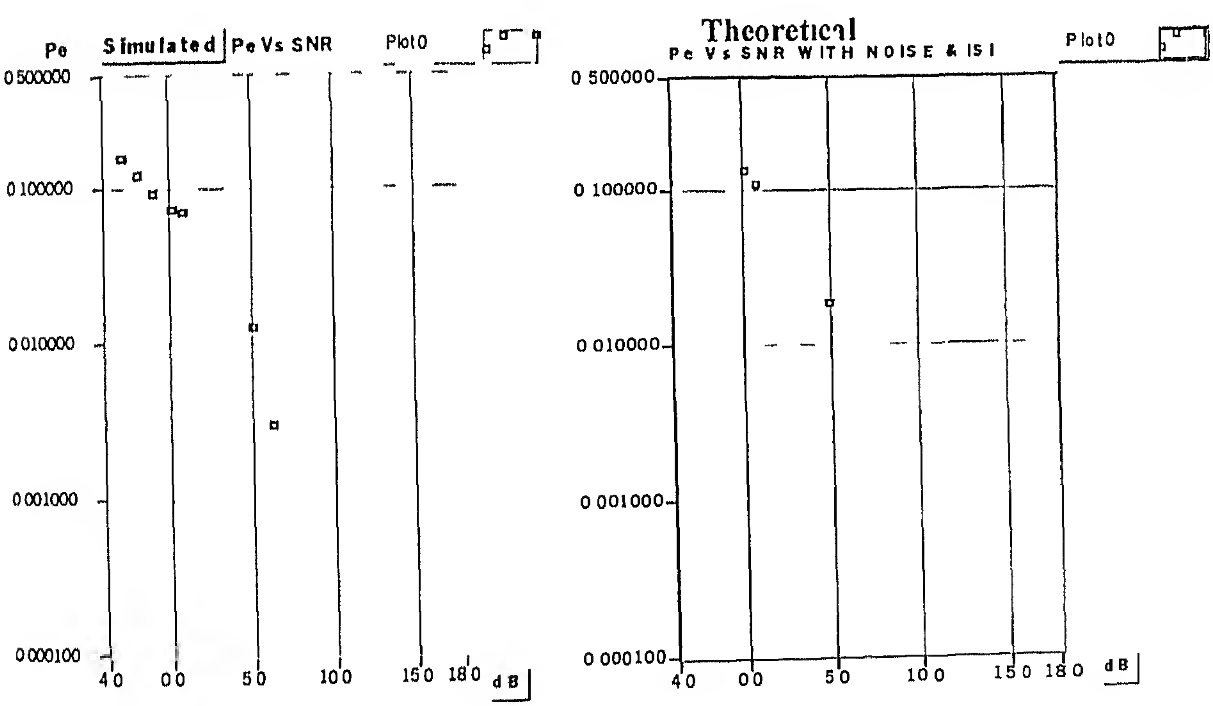
$P_e = P\,[\,|D(i) + \eta_o| > x_o d\,]$ where η_o is the noise sample and $x_o d$ is the distance from threshold

$$P_e = \frac{1}{L^{2k}} \sum_{i}^{L^{k}} \left\{ Q\left[\frac{x_o d - D(i)}{\sigma_n}\right] + Q\left[\frac{x_o d + D(i)}{\sigma_n}\right] \right\}$$

where the noise is gaussian with zero mean and variance σ_n^2 [23]

D(i) has been evaluated for each input sequence to the adaptive equalizer in the adaptive equalizer vi The threshold distance $x_o d$ has been taken as 1 The theoretical Pe has been calculated by averaging over all input sequences The signal amplitude is 1 the detection errors at varying noise variance have been evaluated for a 1000 bit binary input sequence

Simulated graphs of the practical implementation and theoretical implementation are given in plot VIII



Plot VIII

4 5 REGENERATION OF EXTERNAL SIGNAL AFTER PROCESSING

A sinusoidal signal at 1kHz frequency is taken from a function generator through DAQ interface card and DAQ function generator The DAQ function in LabVIEW C Scan is used to scan 15 cycles 100 sample points each at a sampling rate of 10000 from channel 1 of device 1 (function generator) This input signal is quantised and BPSK modulated The BPSK receiver and decoder regenerate this 1KHz signal This reconstructed signal through the DAQ interface using analogue out continuous generation function is given to the DSS 5040 oscilloscope The output signal on CRO is a 1KHz signal, the quantisation steps at discrete levels are visible on the CRO scope in the digital storage mode

CHAPTER 5

CONCLUSION

The simulation system which has been described allows interactive digital simulation of arbitrary point to point digital communication system The software has been structured to allow flexible interactive use by even the most casual user The major application of this communication simulator package will be in support of new development efforts in modulation/coding concepts which are emerging anew in the communication world The various system parameters the graphical representation of the waveforms at any stage in the link which can be displayed with ease give the user a better understanding of a practical system Spectrum analyzers or CRO s are not needed as one can view these waveforms or their spectrum on the LabVIEW displays The general philosophy in developing the software has been that it should be used as an adjunct and not a substitute of the system

LabVIEW with its debugging features the graphical support and its interface for data acquisition makes it a very suitable choice for using it for simulation The other simulation packages like simulink and higher level languages like FORTRAN could also have been used but the basic advantage of a graphical interface and data acquisition would not be present The various data operations plots/charts along with the various functions and library of VIs in LabVIEW makes the simulation job much more attractive and easier Simulation in LabVIEW takes one closer to the actual hardware implementation of the system in hardware

The simulation results discussed in chapter 4 do concur with the results obtained in theory [10 13 14 22] The BER plots would have been closer approximations had a larger data sequence been considered and averaging over a larger number of runs been carried out The processor limitation can be abstained by doing these simulation runs over a main frame

Another major advantage of the simulation package is the ease with which it can be updated to incorporate enhanced capabilities or to add new modulation

schemes Modules of these new modulation schemes can be developed and easily be accommodated The ability to conveniently expand system capabilities will ensure that this system will provide a useful tool in support of these efforts

SUGGESTIONS FOR FURTHER WORK

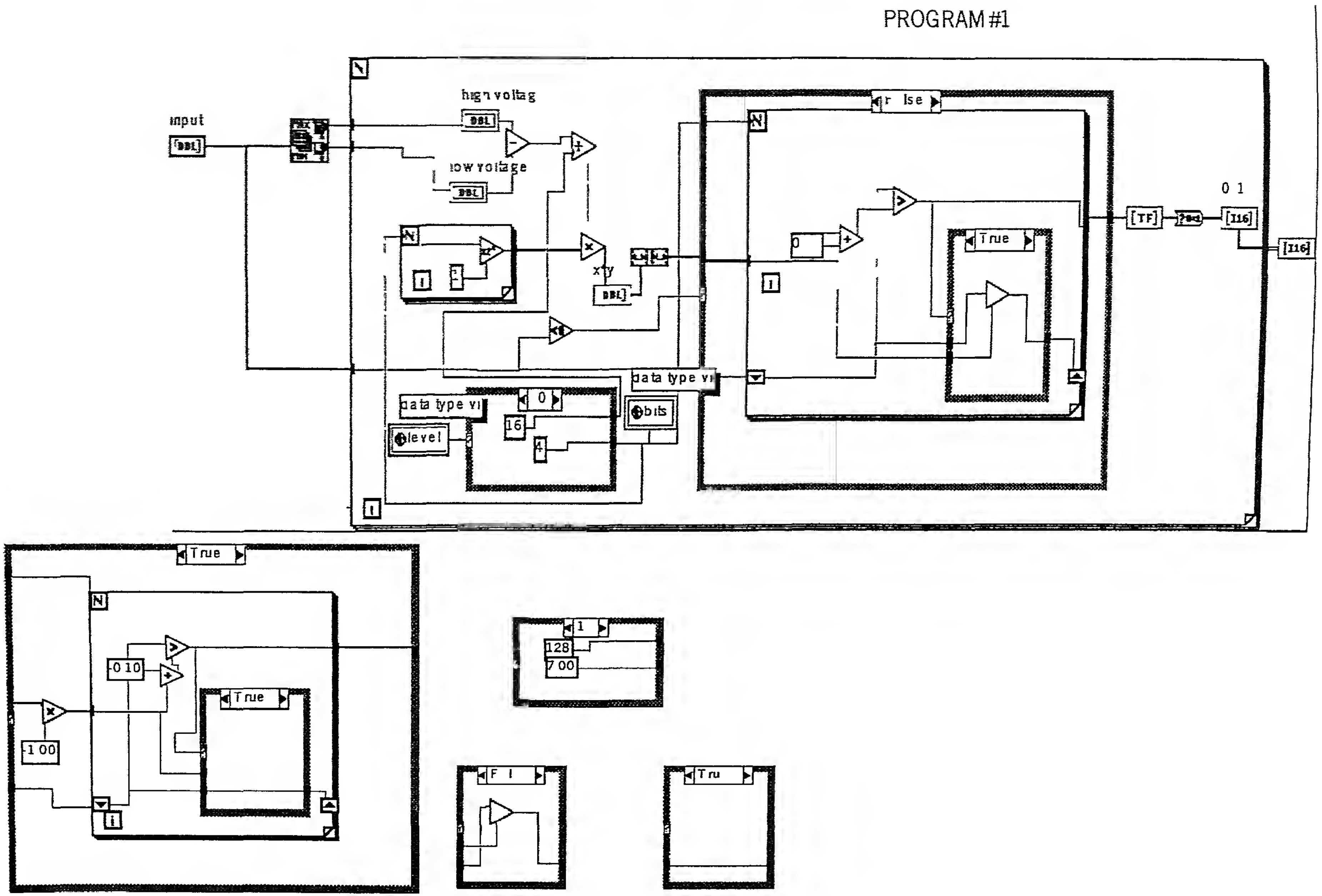
In Section 4 5 it has been indicated that real time signal processing is possible using the DAQ interface If a dedicated host computing machine with a faster processor were used it would be possible to simulate a real time system A transmitter and receiver model could be developed using some modulation format to simulate and recover the transmitted data A real time baseband communication system could be developed where the transmitter could be on one PC and the receiver on another

The modulation format which have been developed are the basic ones modern modulation techniques used like TCM OFDM could be incorporated Multiple access techniques like TDMA FDMA can also be included as simulation functions Carrier and bit synchronization can be included in the simulation feature by designing a 2nd order PLL module Random error correcting codes like convolution codes BCH codes and other such schemes can also be provided

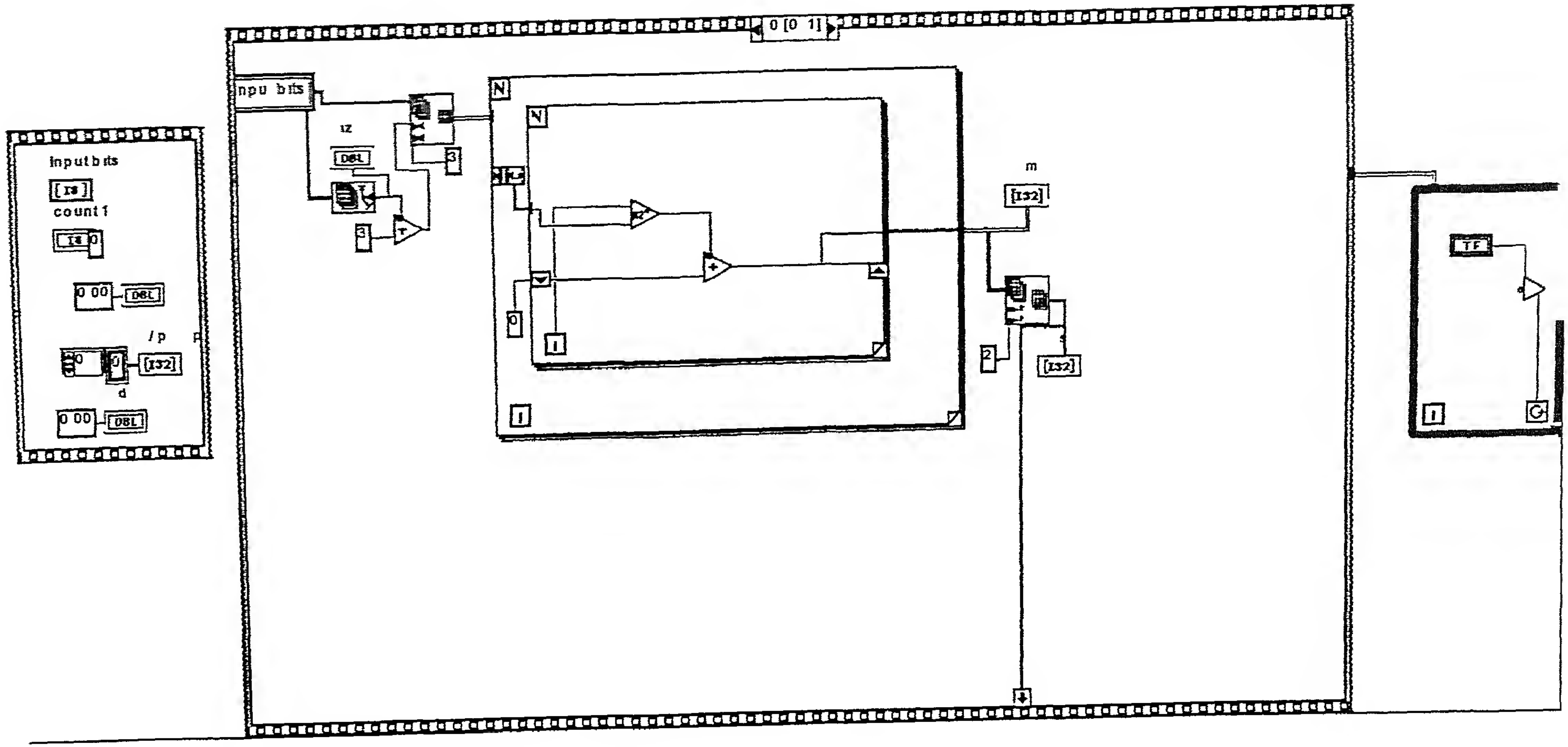
The channel assumed is a Gaussian channel A practical channel would consist of a fading multipath (dispersive) channel along with the AWGN channel A fading channel could be incorporated in the simulation The distortions due to narrow band filtering can also be included as a function of the transmitter In the baseband schemes raised cosine filters at the transmitter and receiver could be implemented The equalizers which have been designed for 9 taps could be extended to 65 taps (gain coefficient in practical systems) and more discrete channel model may be incorporated in simulation

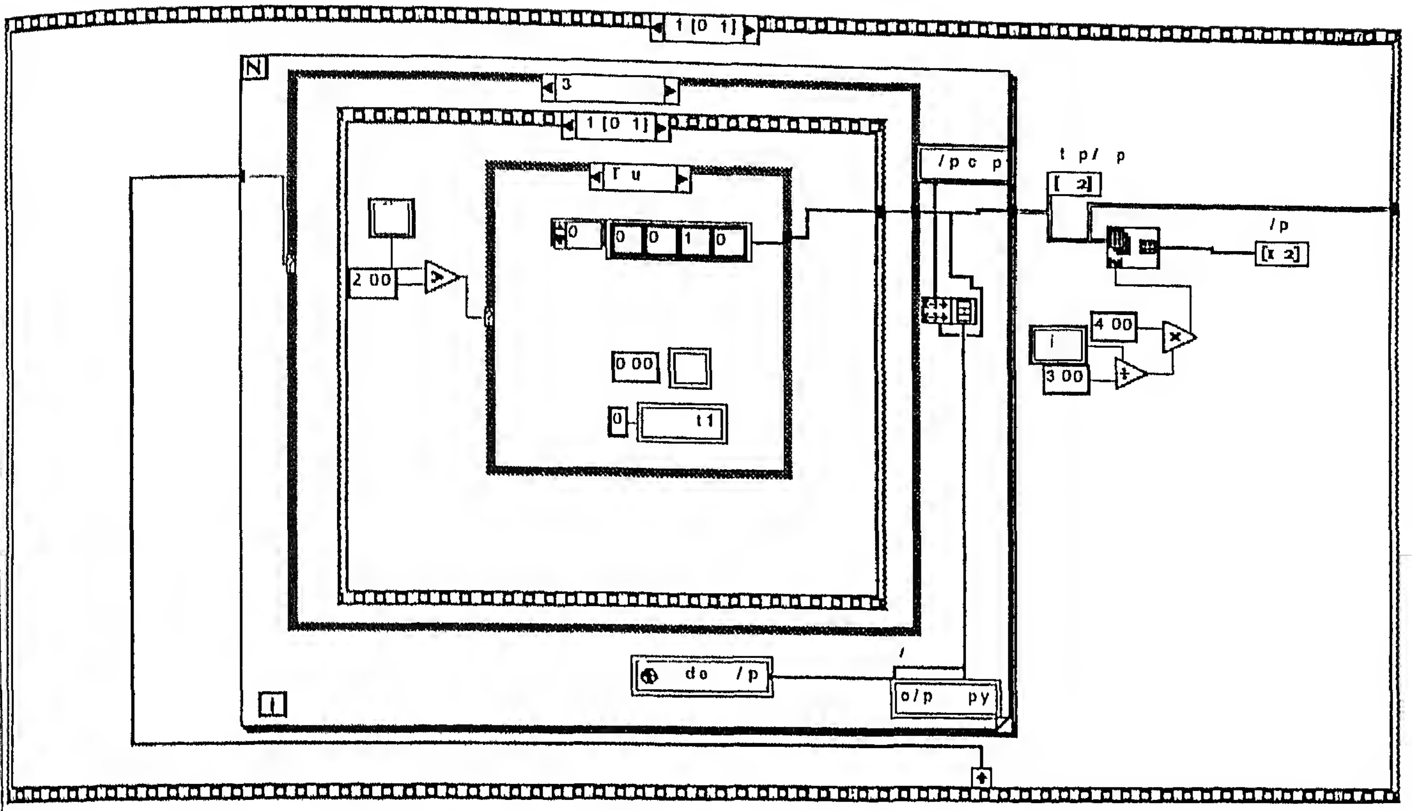
APPENDIX

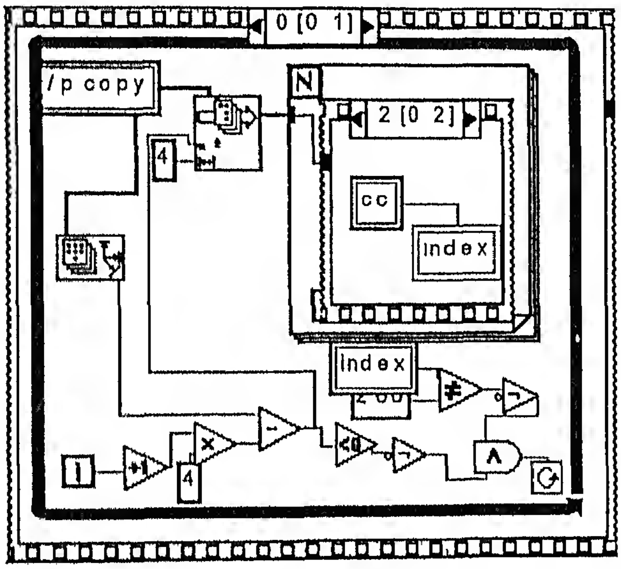
128 LEVEL LINEAR QUANTISATION (program #1)



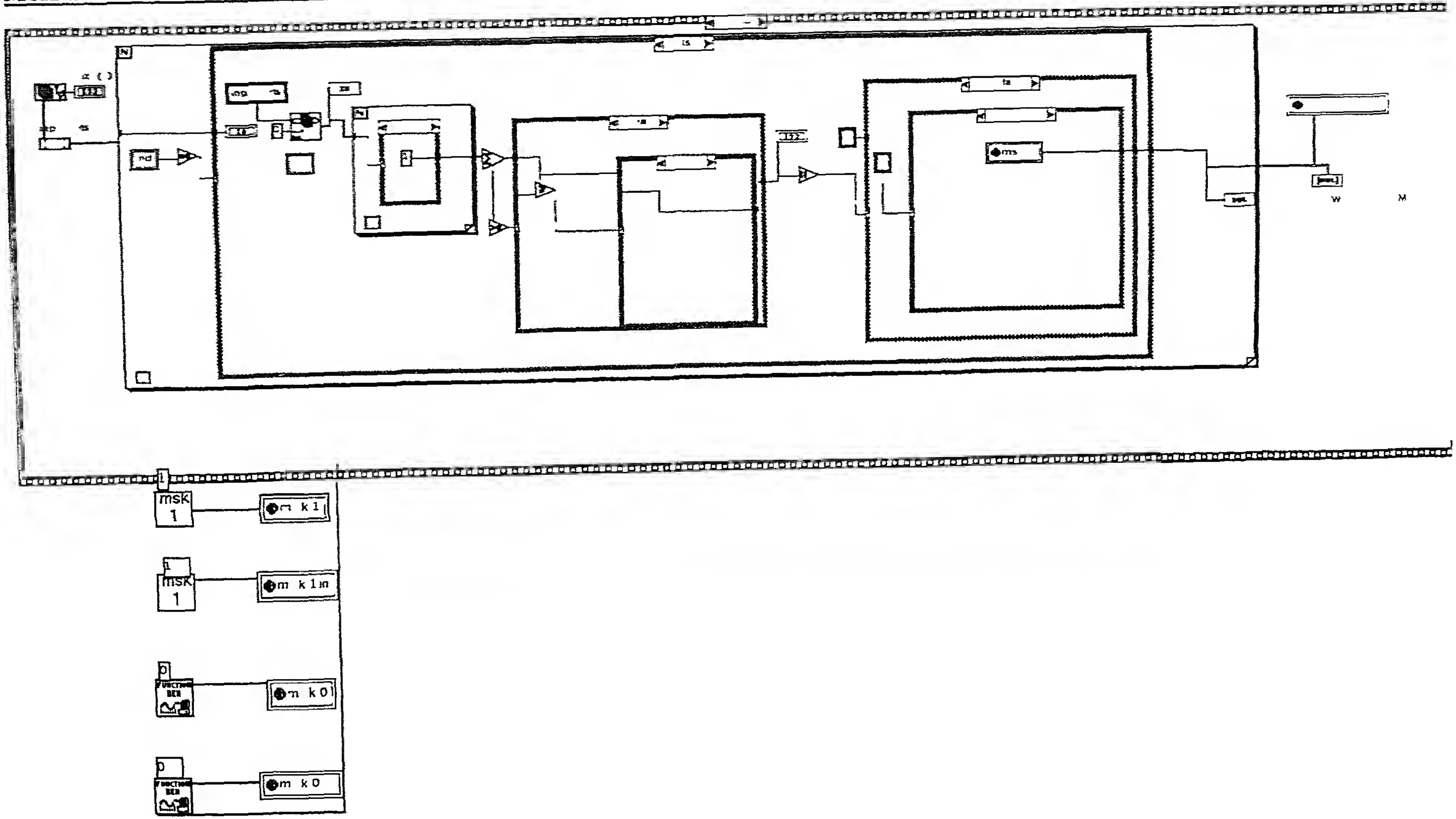
3 BINARY 4 BINARY (PROGRAM # 2)



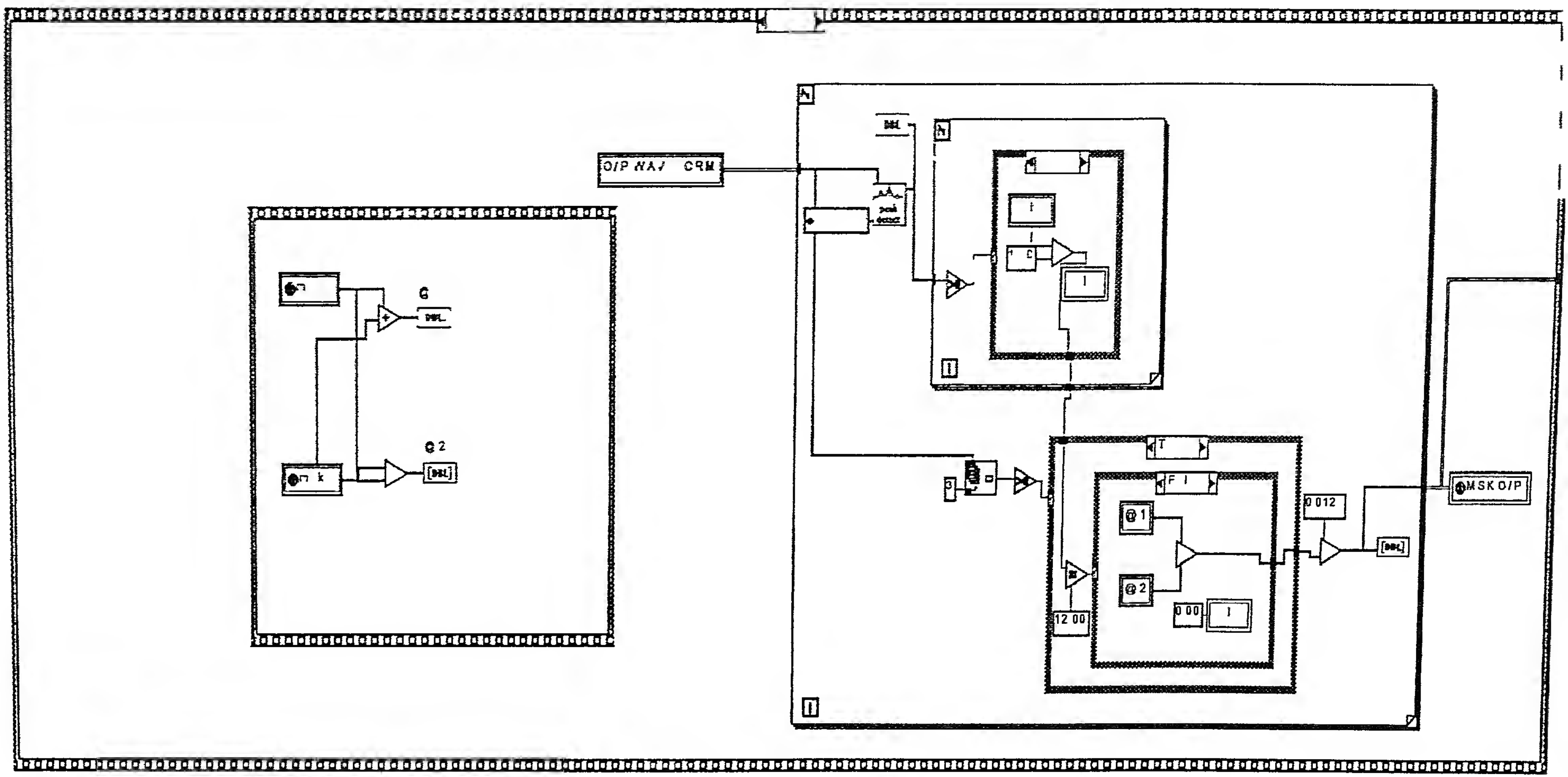




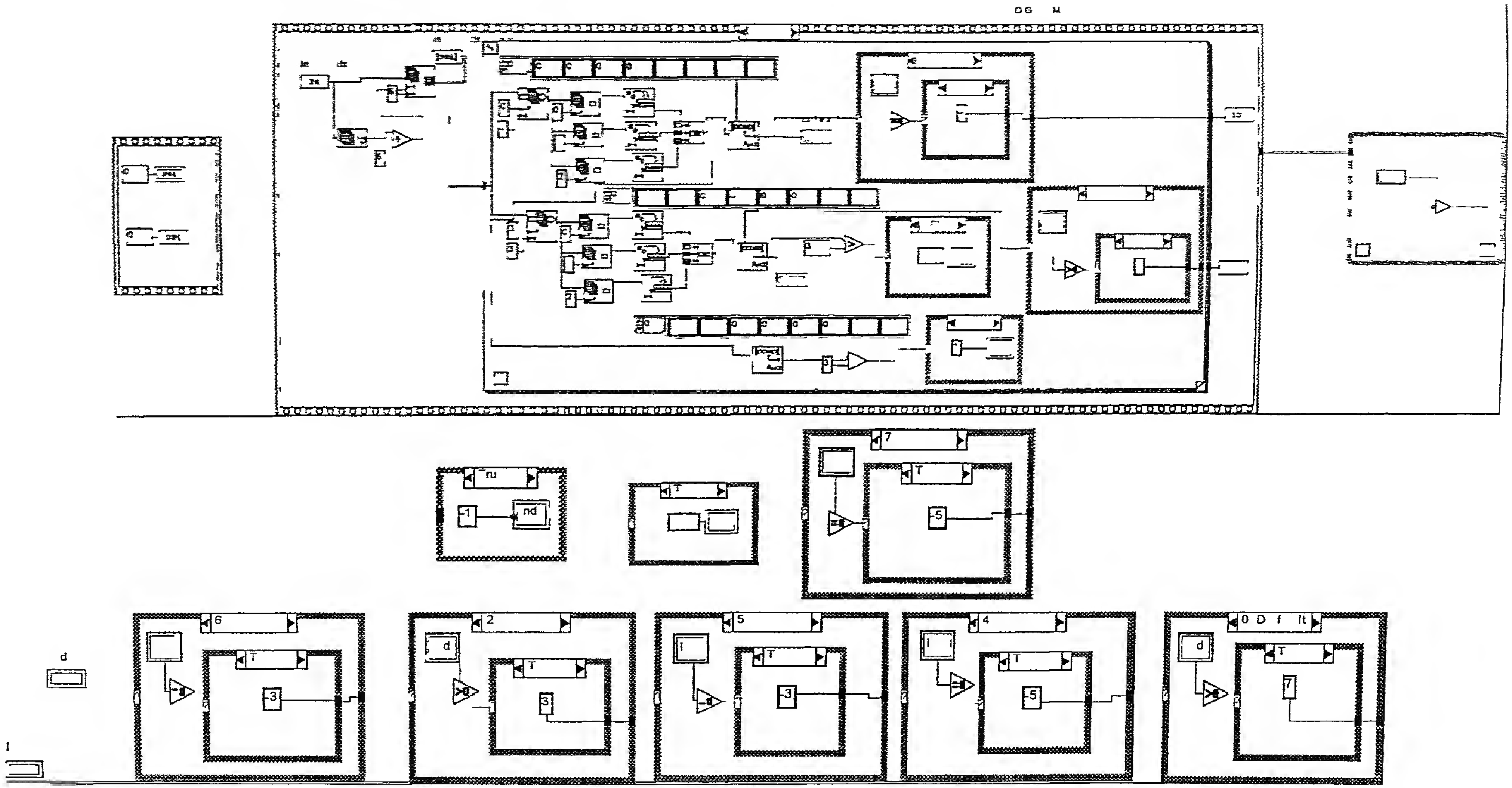
MSK TRANSMITTER (PROGRAM#3)



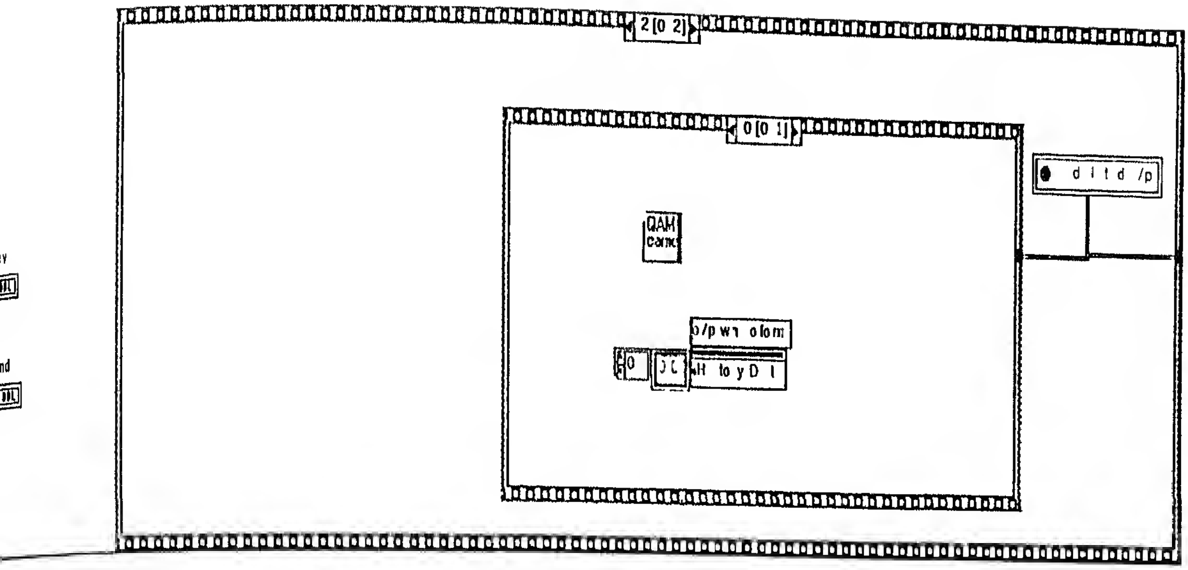
MSK TRANSMITTER

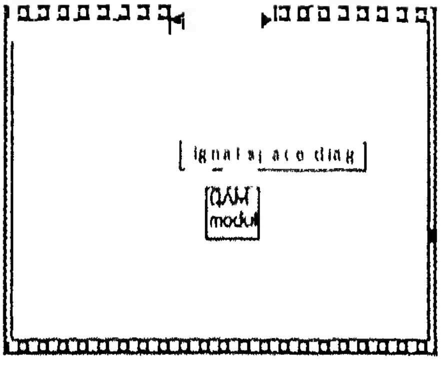


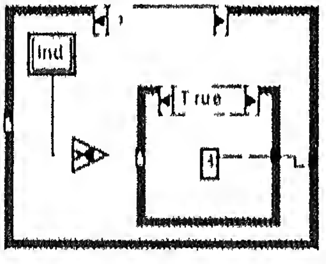
64 QAM TRANSMITTER (PROGRAM # 4)

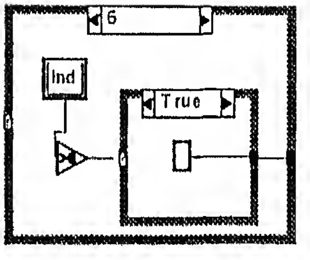


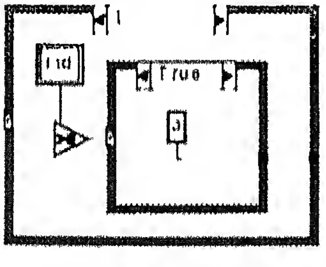
64 QAM TRANSMITTER

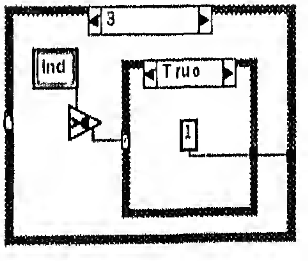




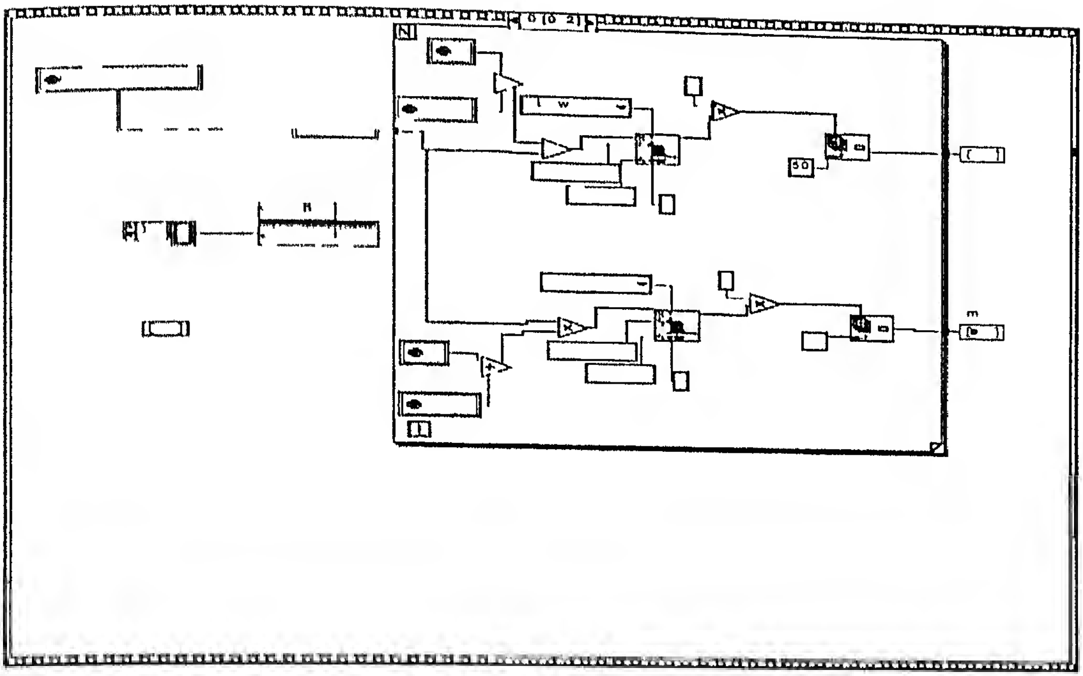


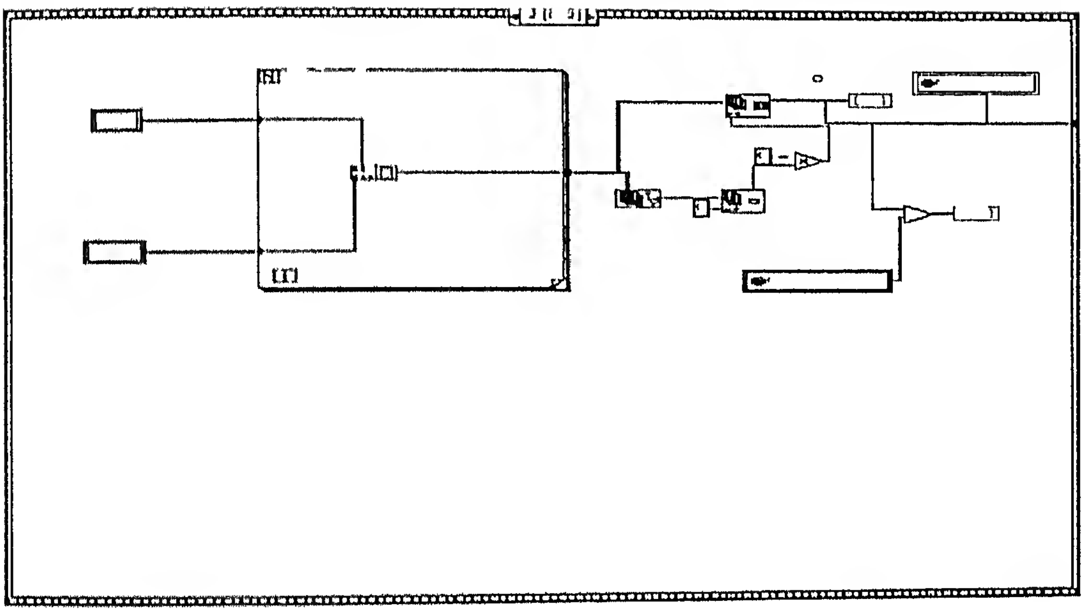




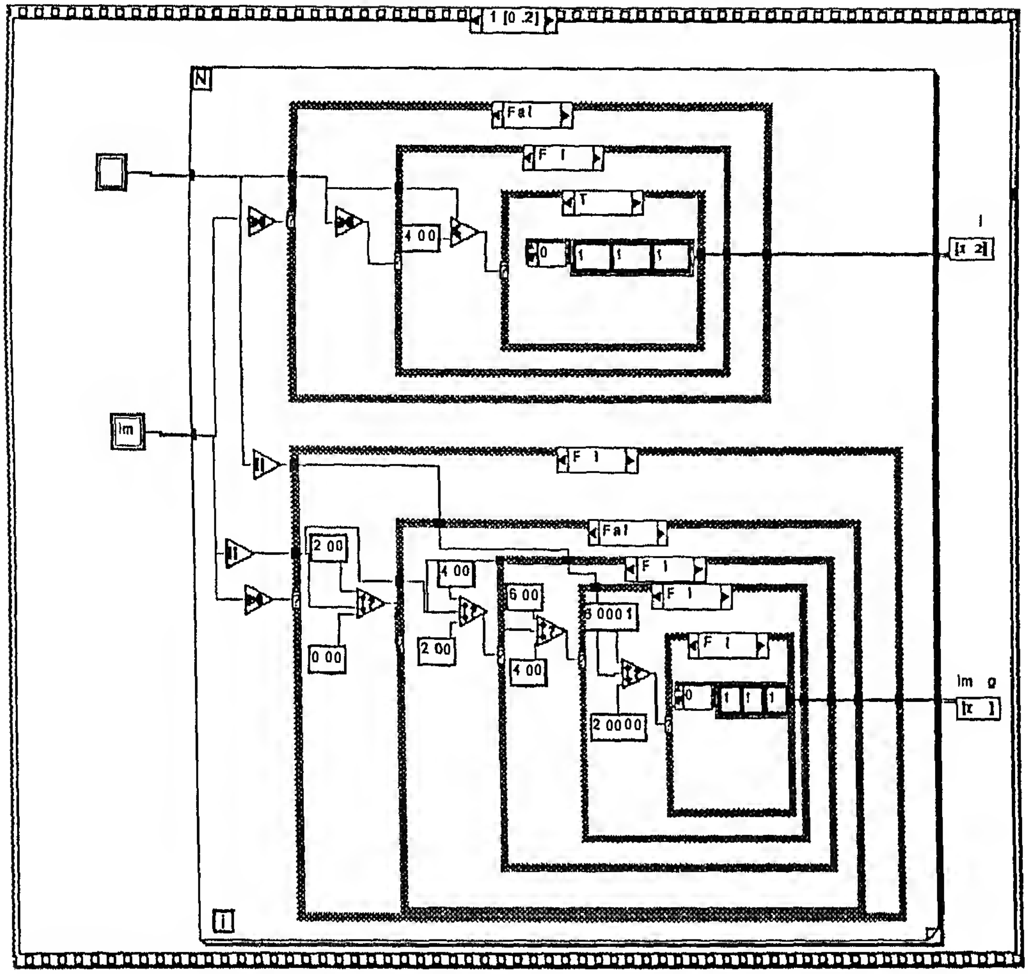


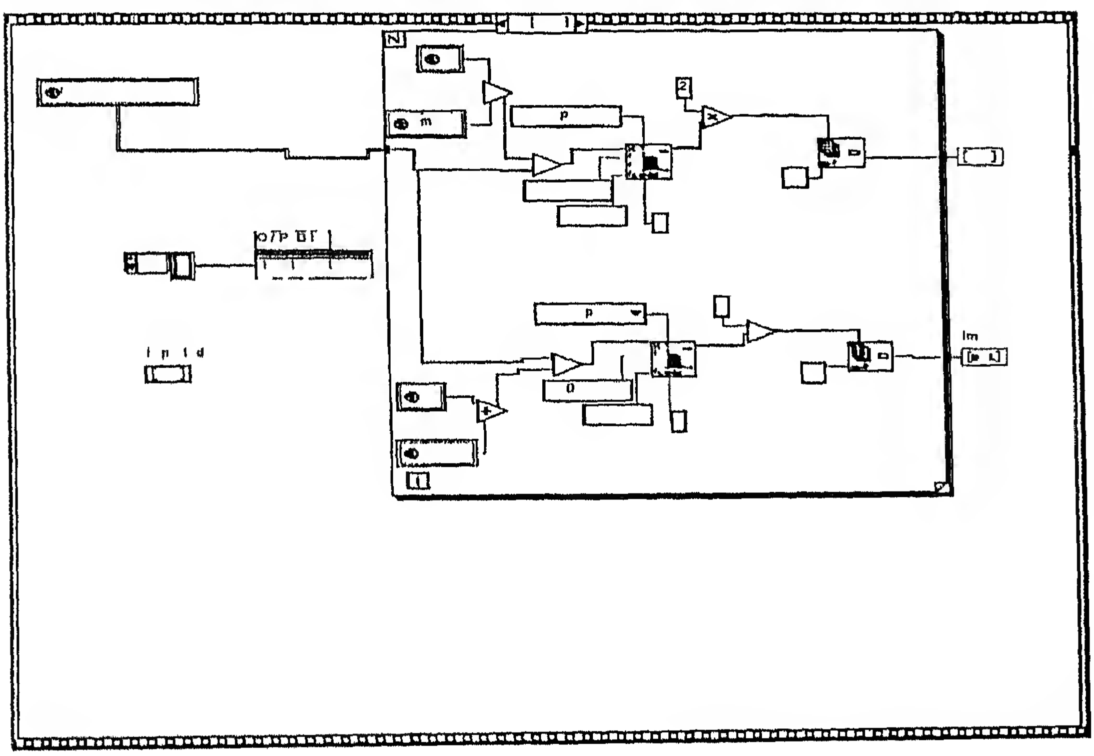
64 QAM RECEIVER (PROGRAM # 5)



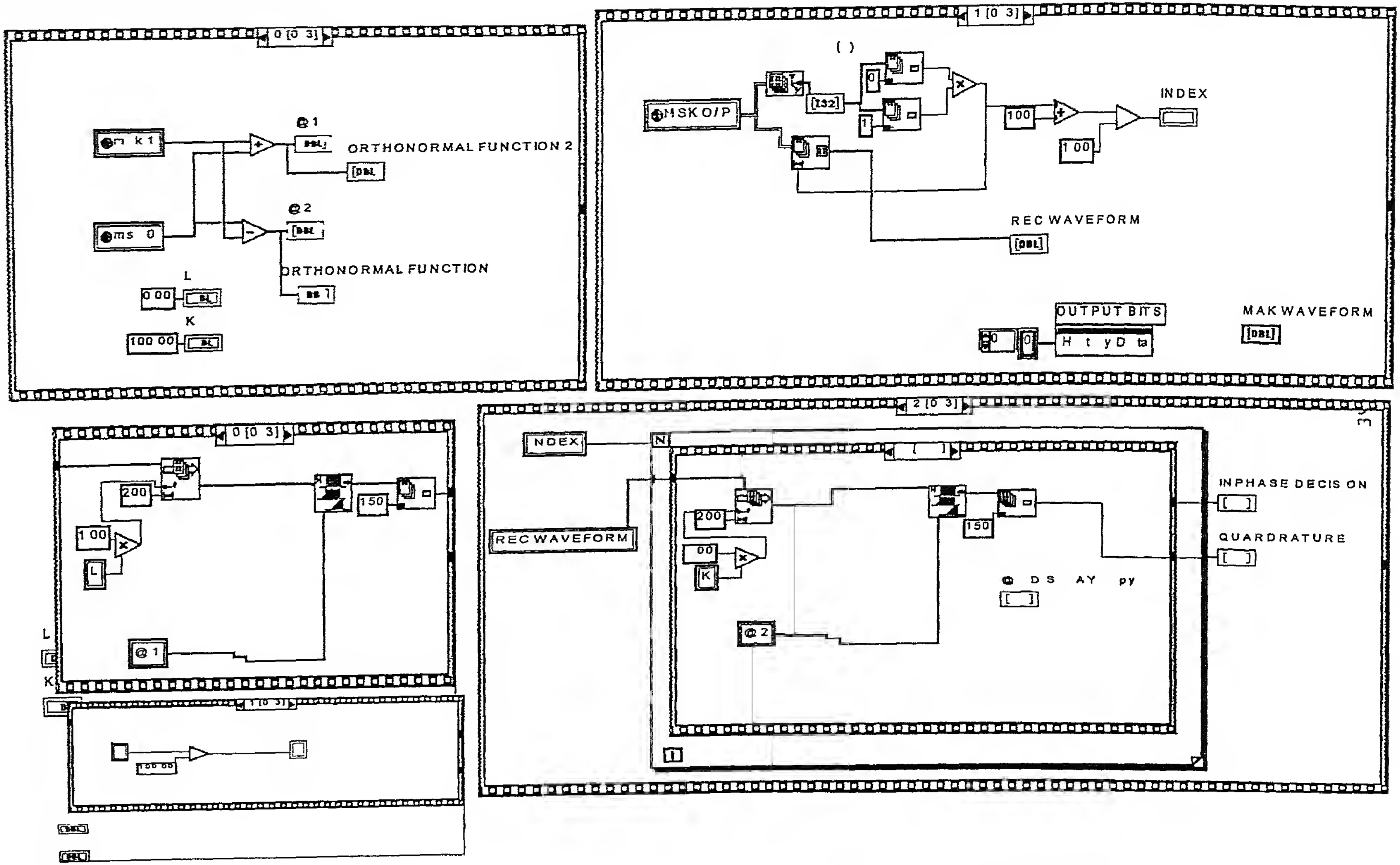


64 QAM RECEIVER

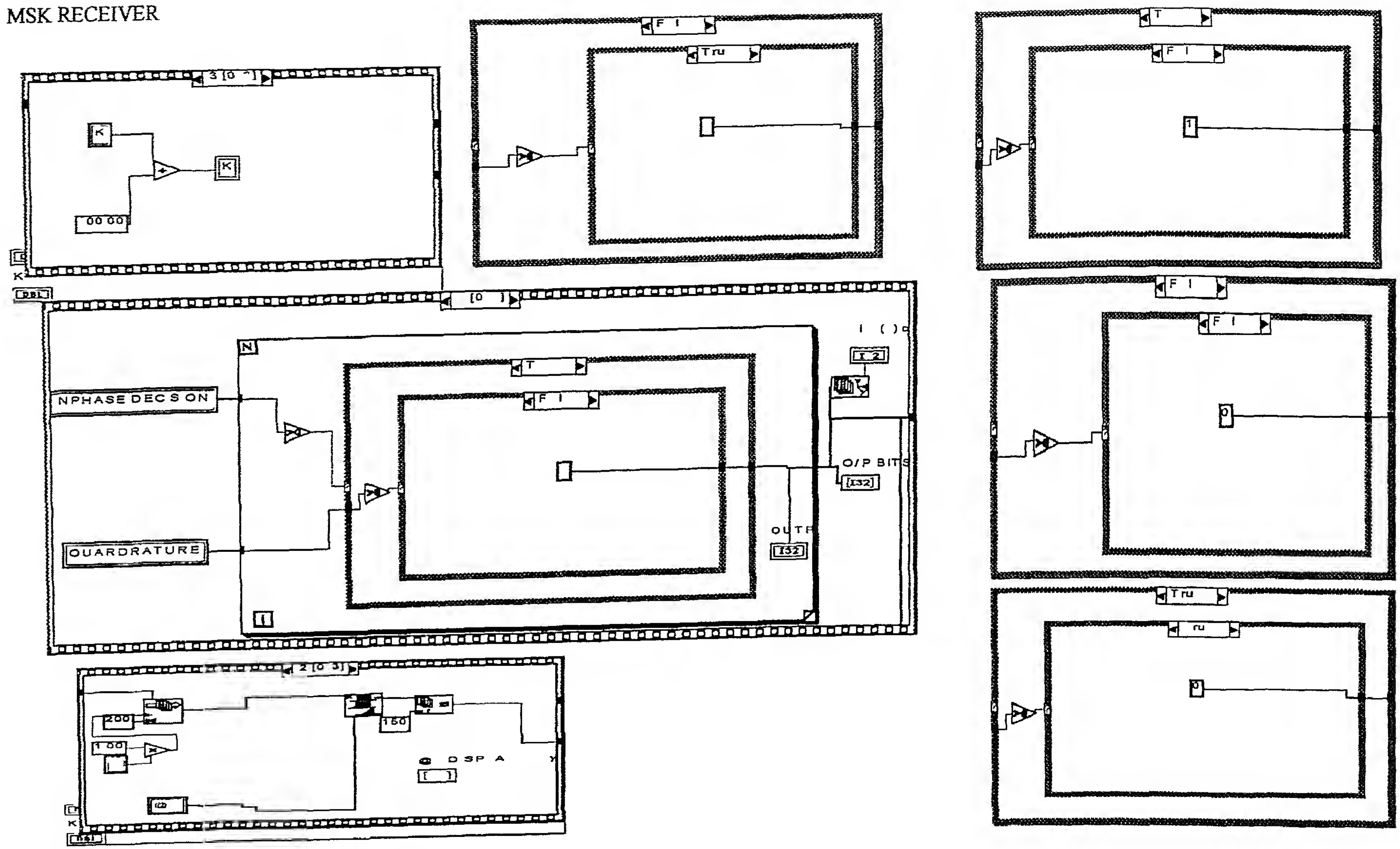




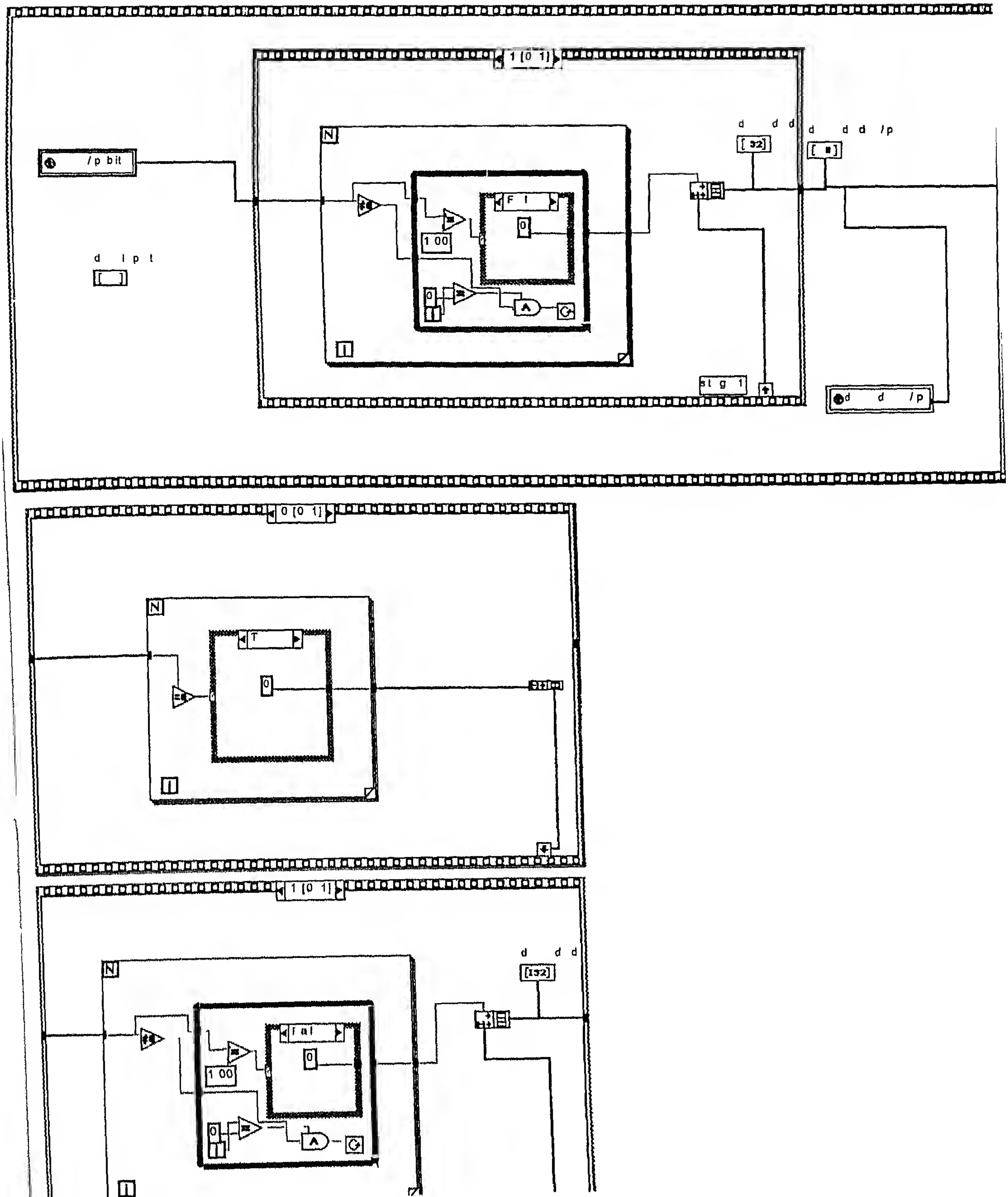
MSK RECEIVER (PROGRAM # 6)



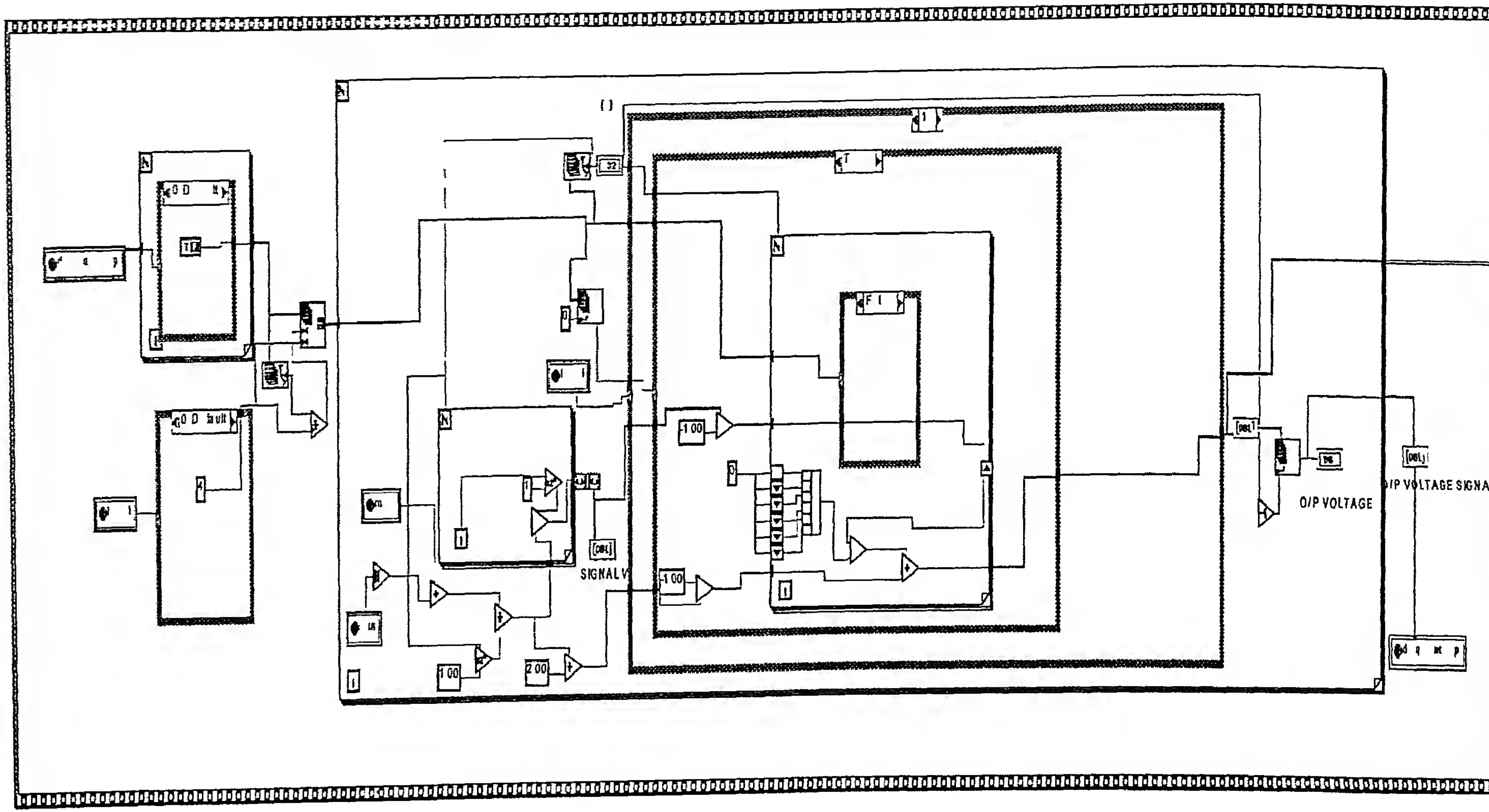
MSK RECEIVER

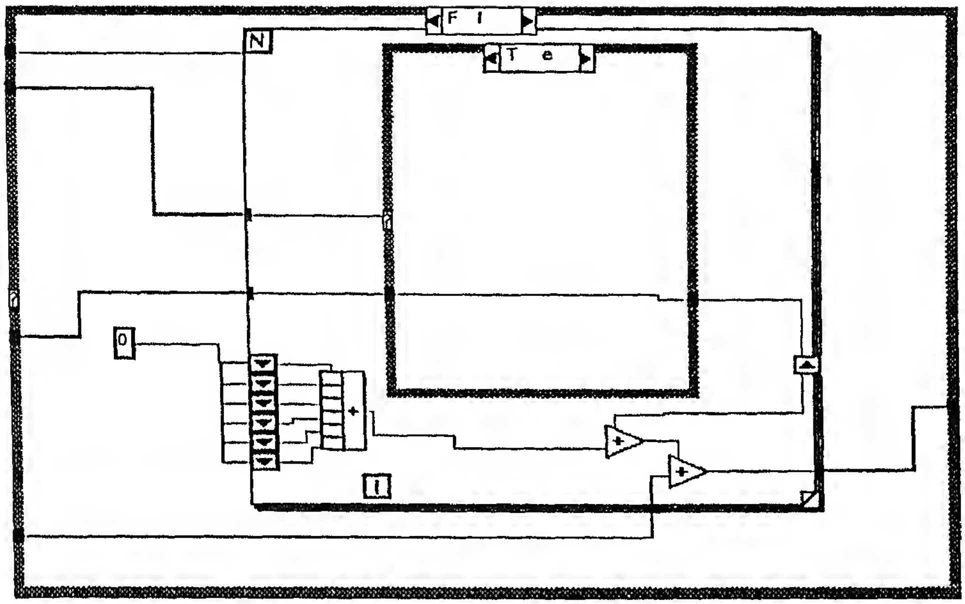


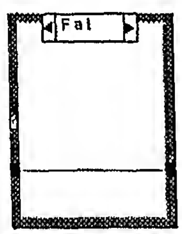
AMI DECODER (PROGRAM #7)

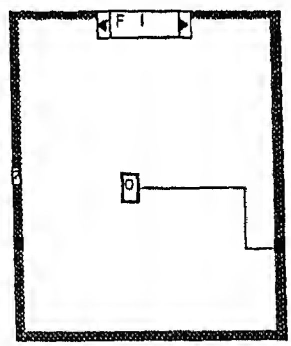


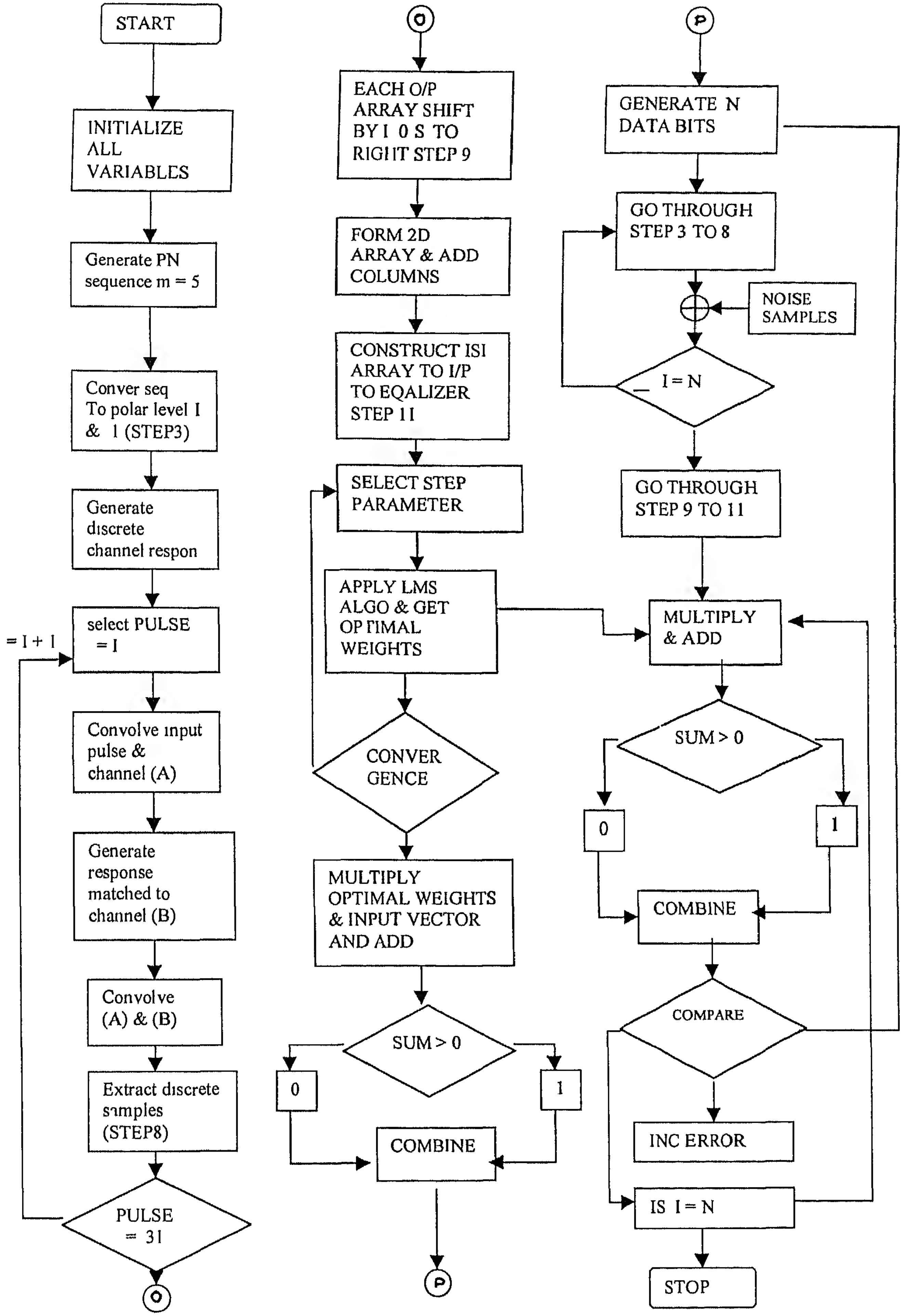
128 LEVEL DECODER (PROGRAM # 8)



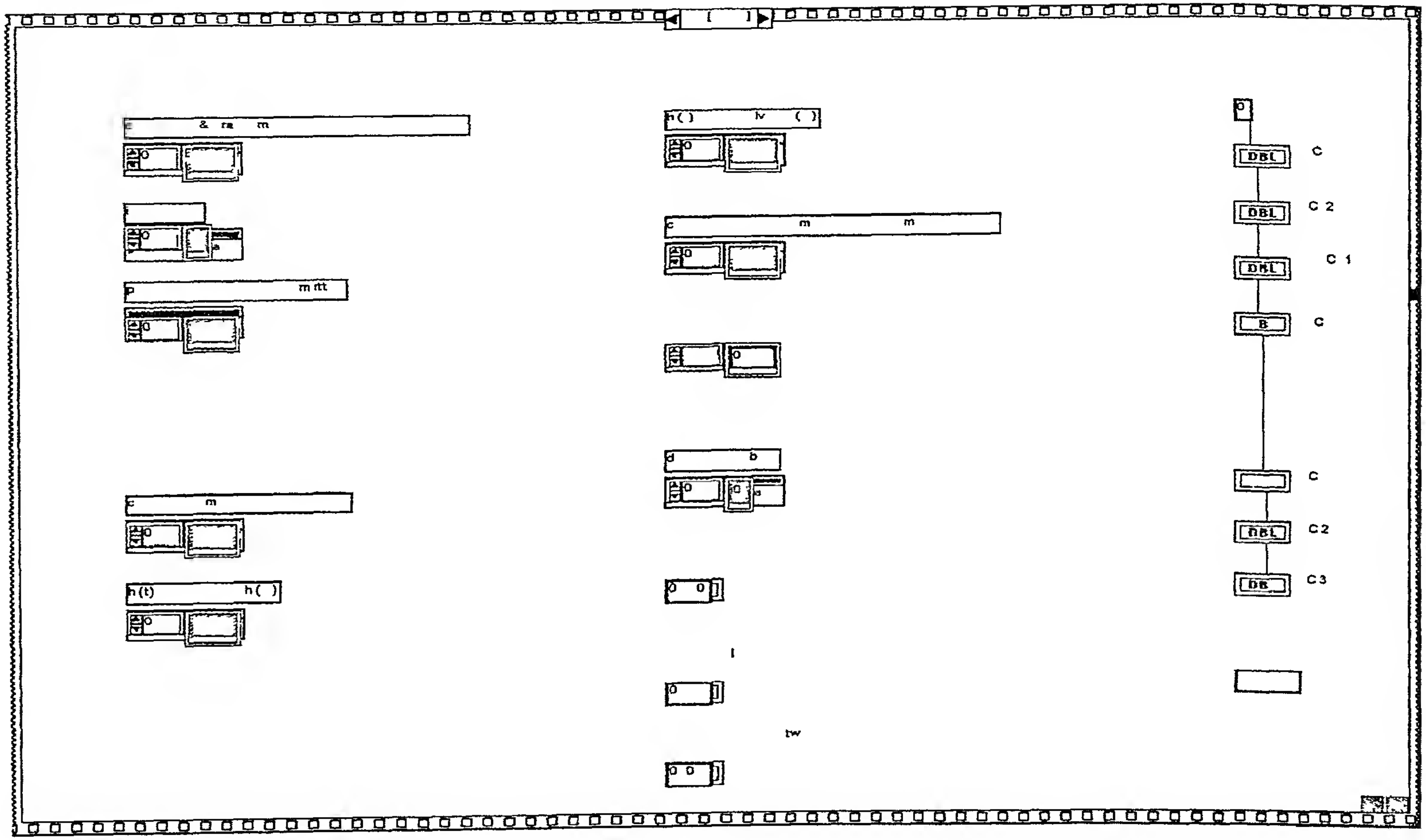




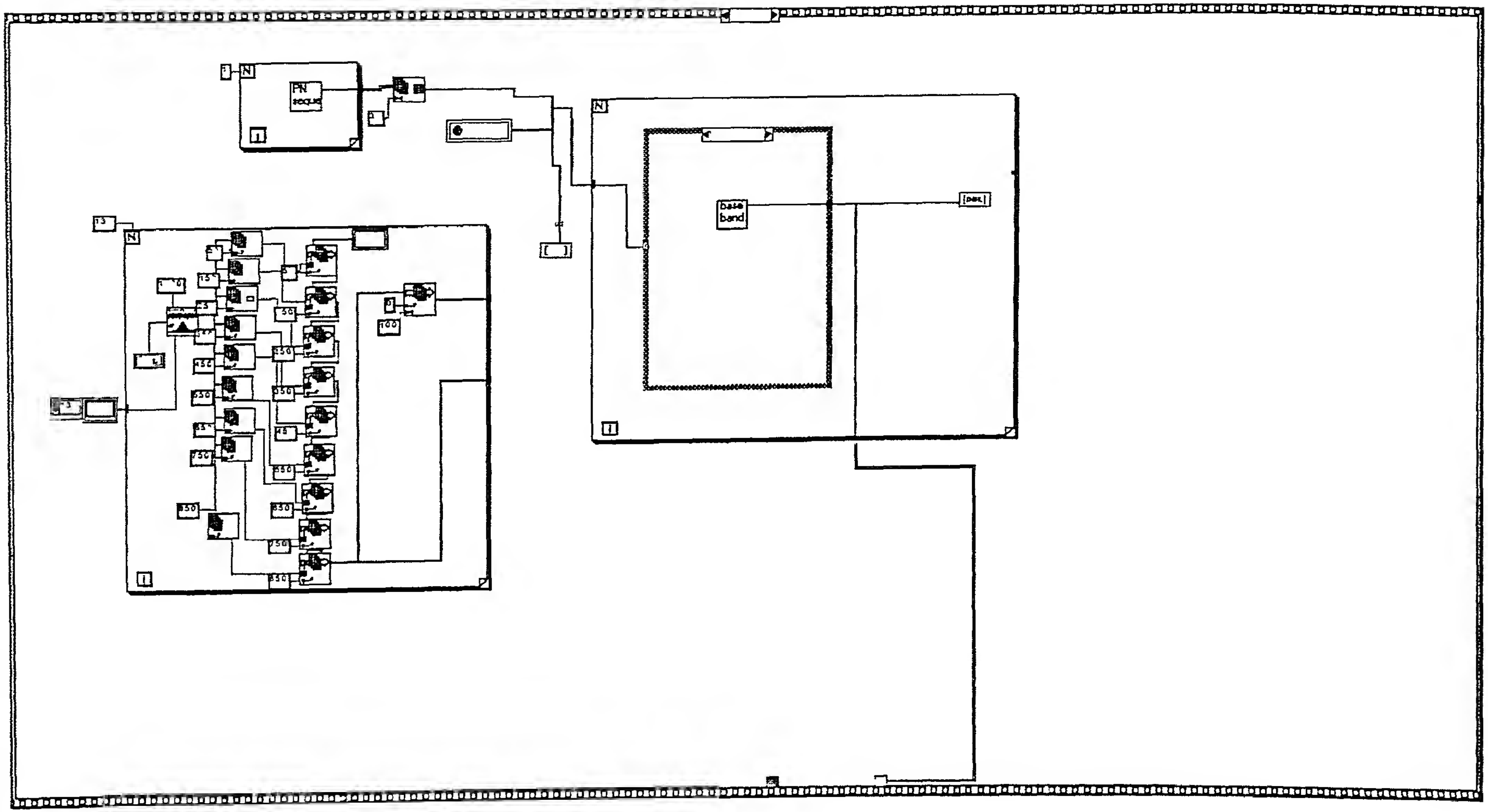



START
INITIALIZE ALL VARIABLES
Generate PN sequence m = 5
Conver seq To polar level I & I (STEP3)
Generate discrete channel respon
select PULSE = I
= I + 1
Convolve input pulse & channel (A)
Generate response matched to channel (B)
Convolve (A) & (B)
Extract discrete samples (STEP8)
PULSE = 31
O
EACH O/P ARRAY SHIFT BY I 0 S TO RIGHT STEP 9
FORM 2D ARRAY & ADD COLUMNS
CONSTRUCT ISI ARRAY TO I/P TO EQALIZER STEP 11
SELECT STEP PARAMETER
APPLY LMS ALGO & GET OPTIMAL WEIGHTS
CONVER GENCE
MULTIPLY OPTIMAL WEIGHTS & INPUT VECTOR AND ADD
SUM > 0
0
1
COMBINE
P
GENERATE N DATA BITS
GO THROUGH STEP 3 TO 8
NOISE SAMPLES
I = N
GO THROUGH STEP 9 TO 11
MULTIPLY & ADD
SUM > 0
0
1
COMBINE
COMPARE
INC ERROR
IS I = N
STOP

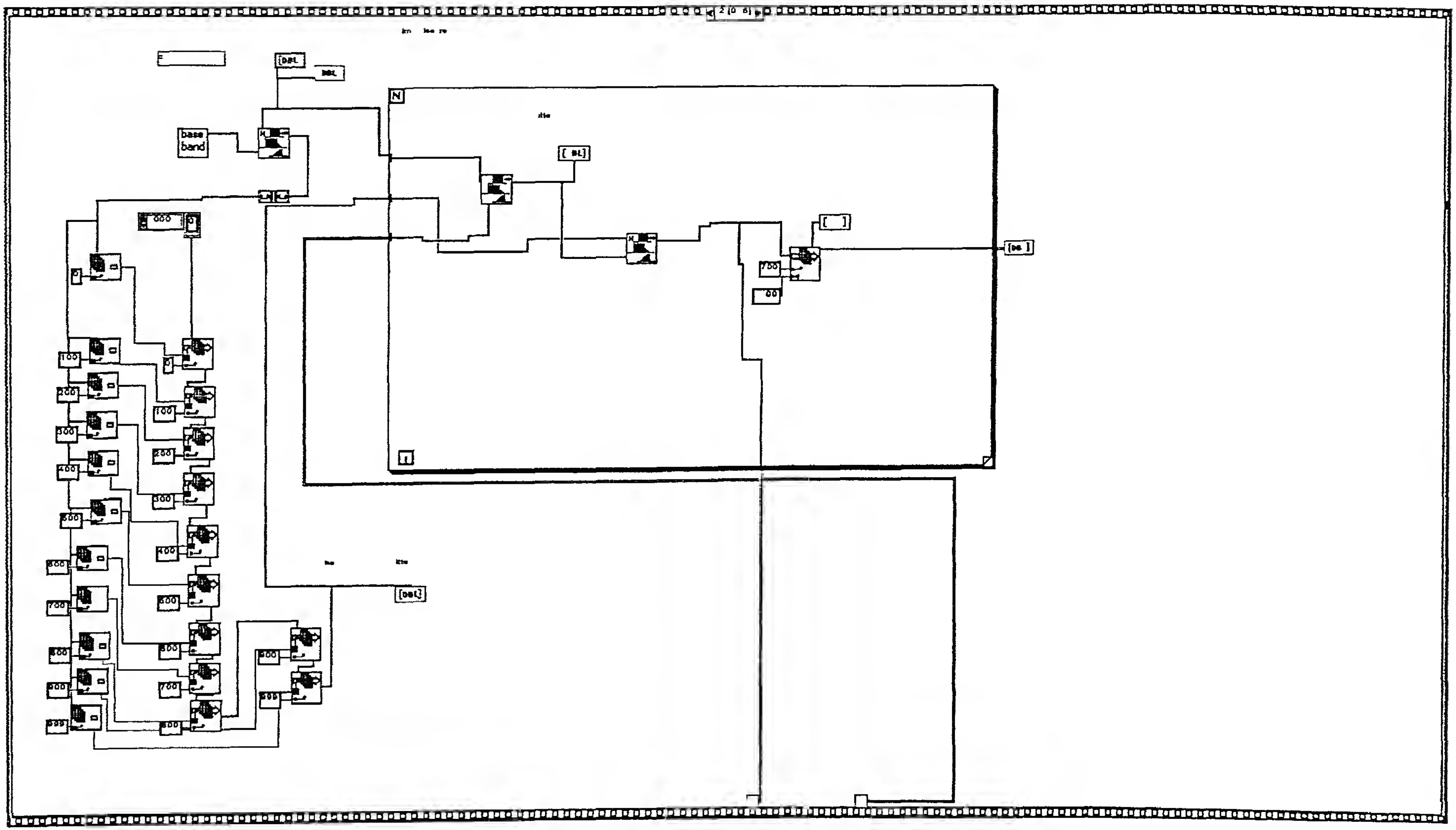
PRESET V1 (PROGRAM # 9)



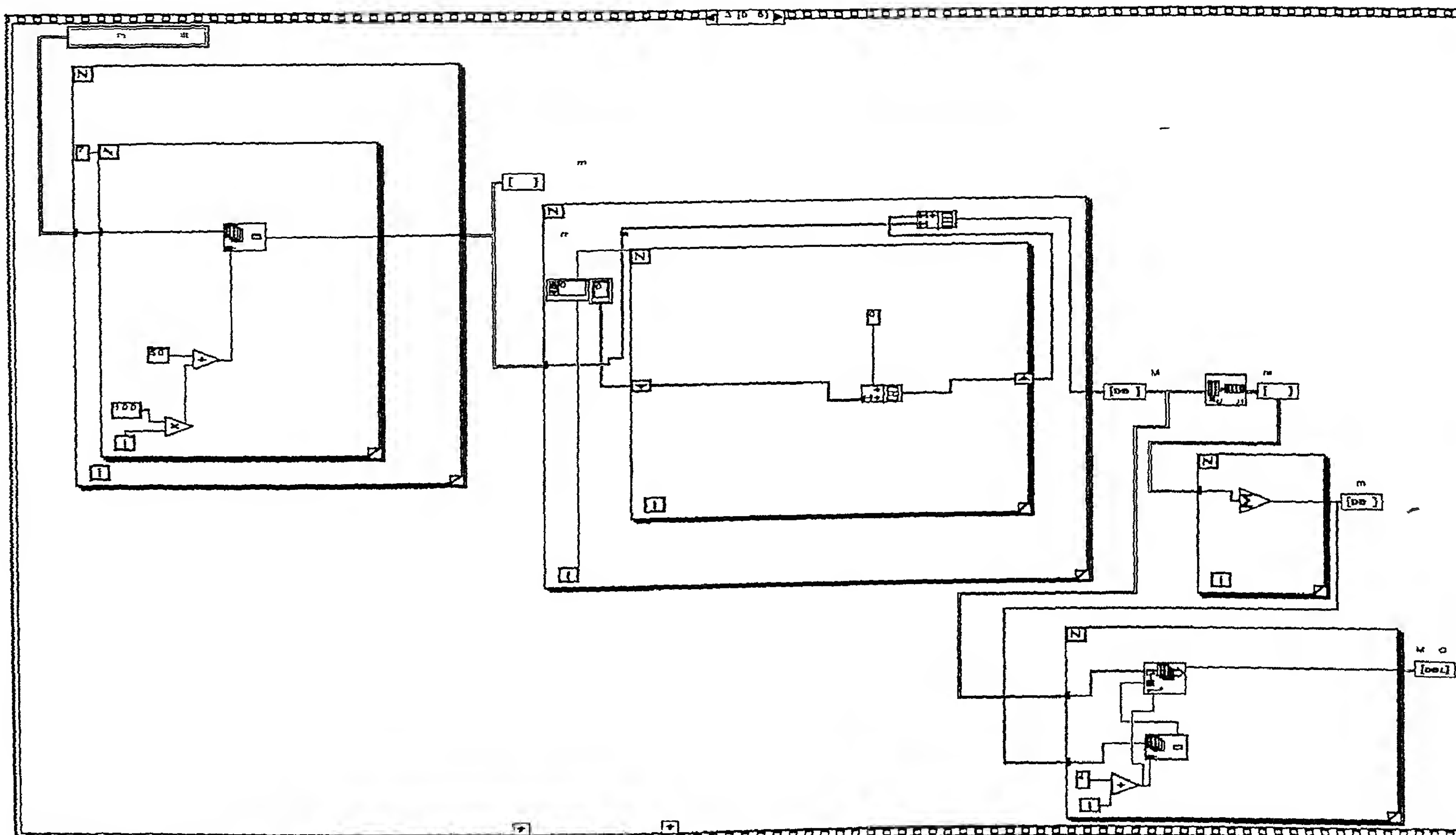
PRESET VI



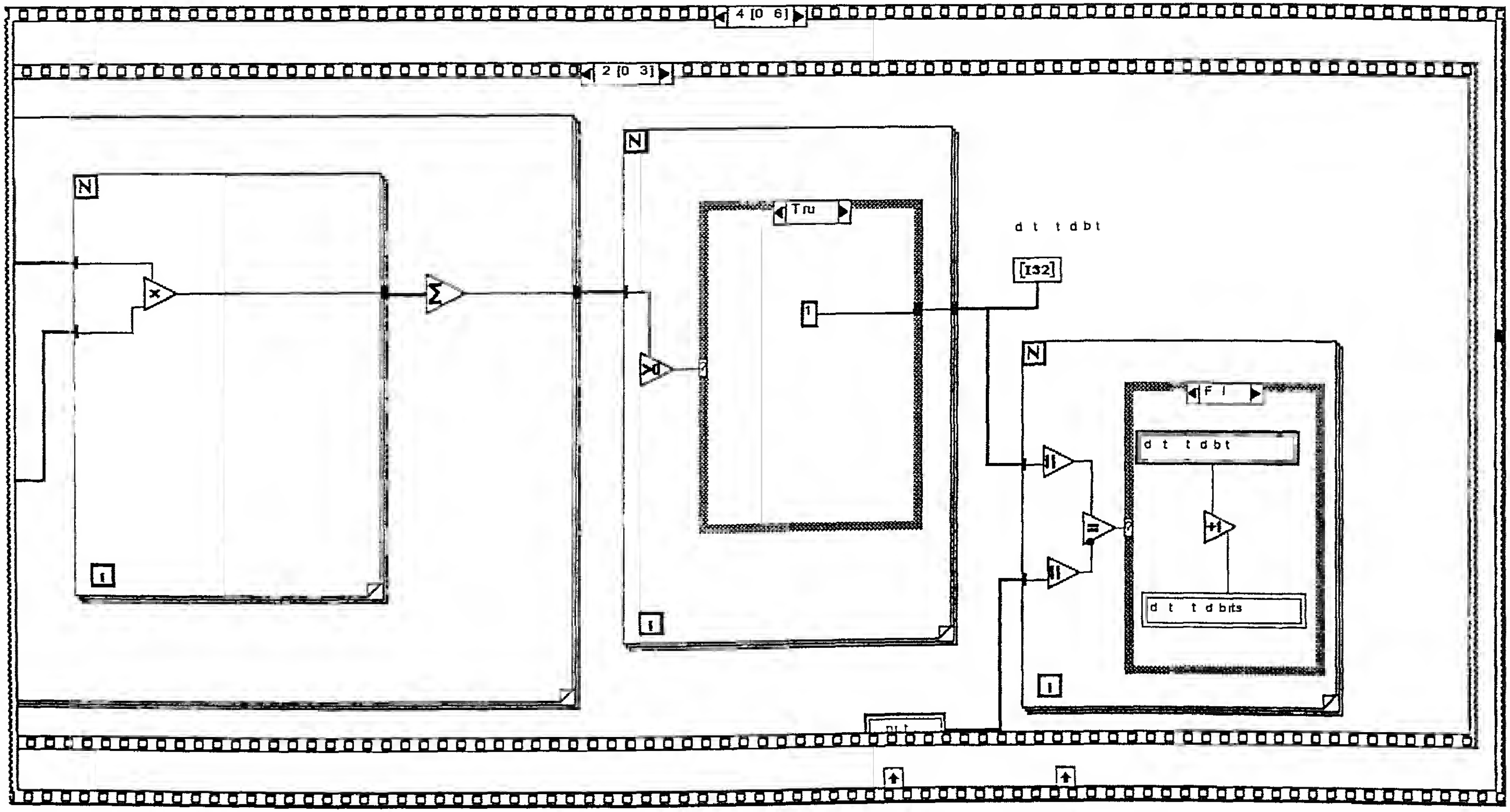
PRESET VI



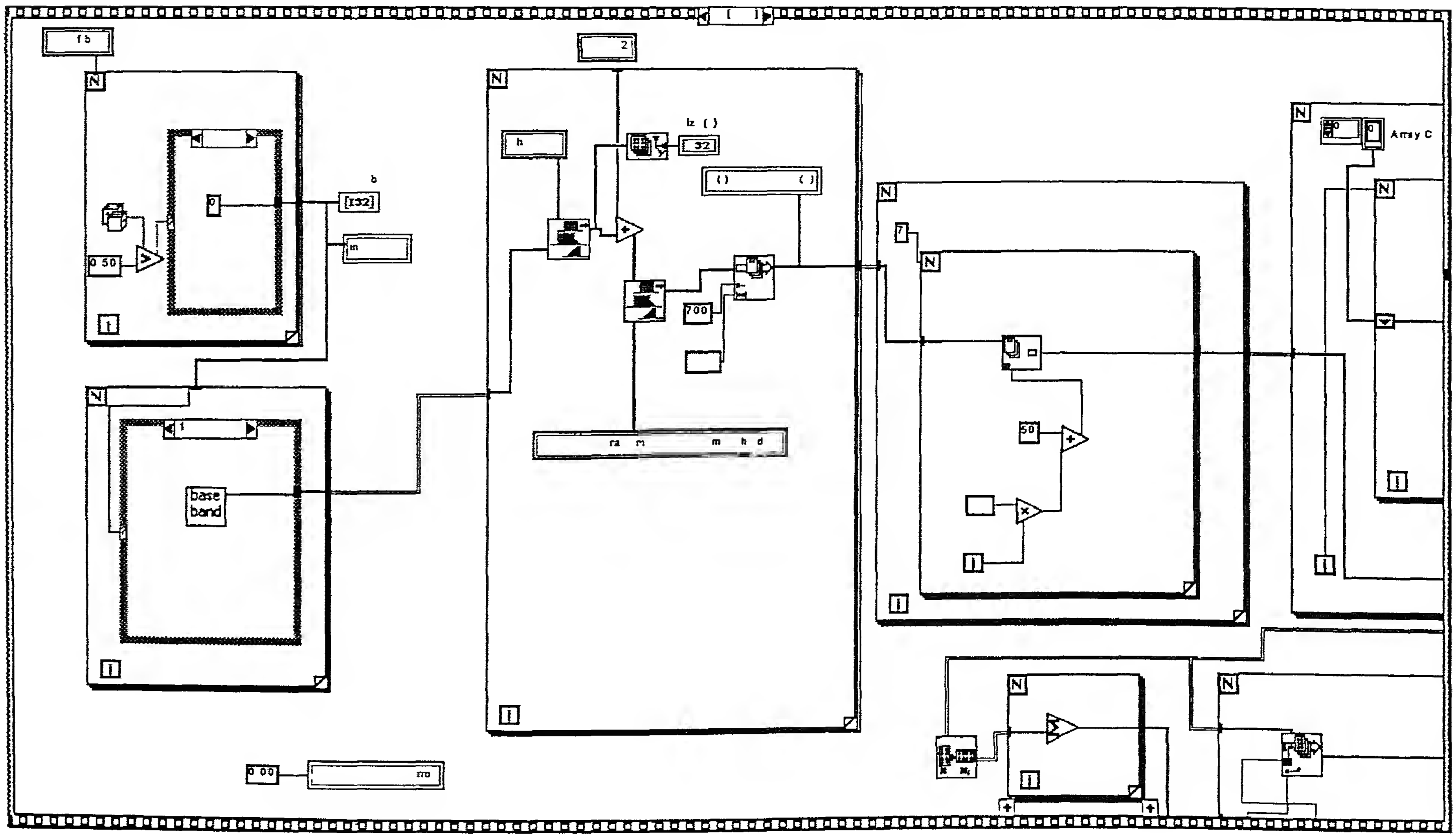
PRESET VI



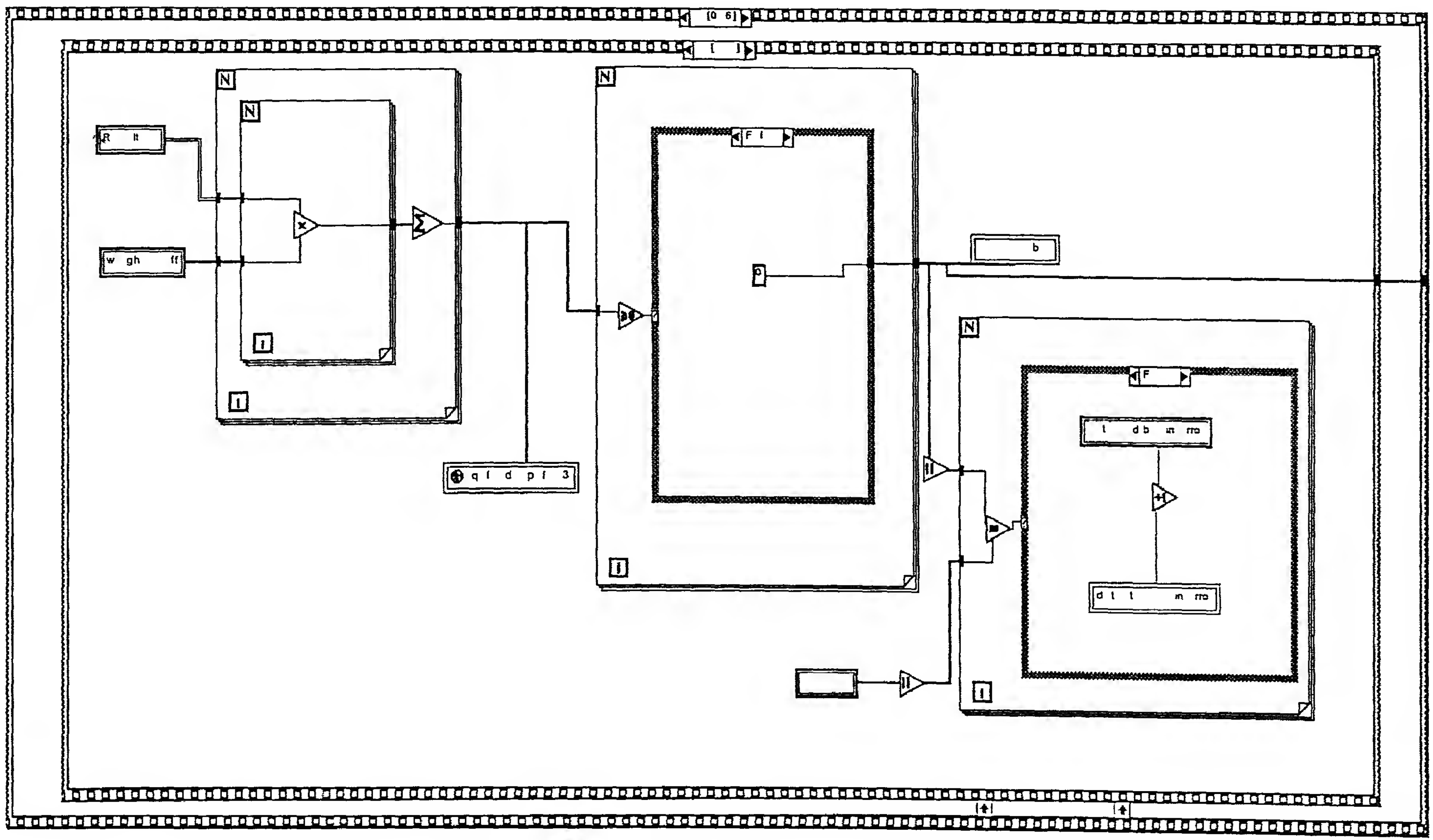
PRESET VI

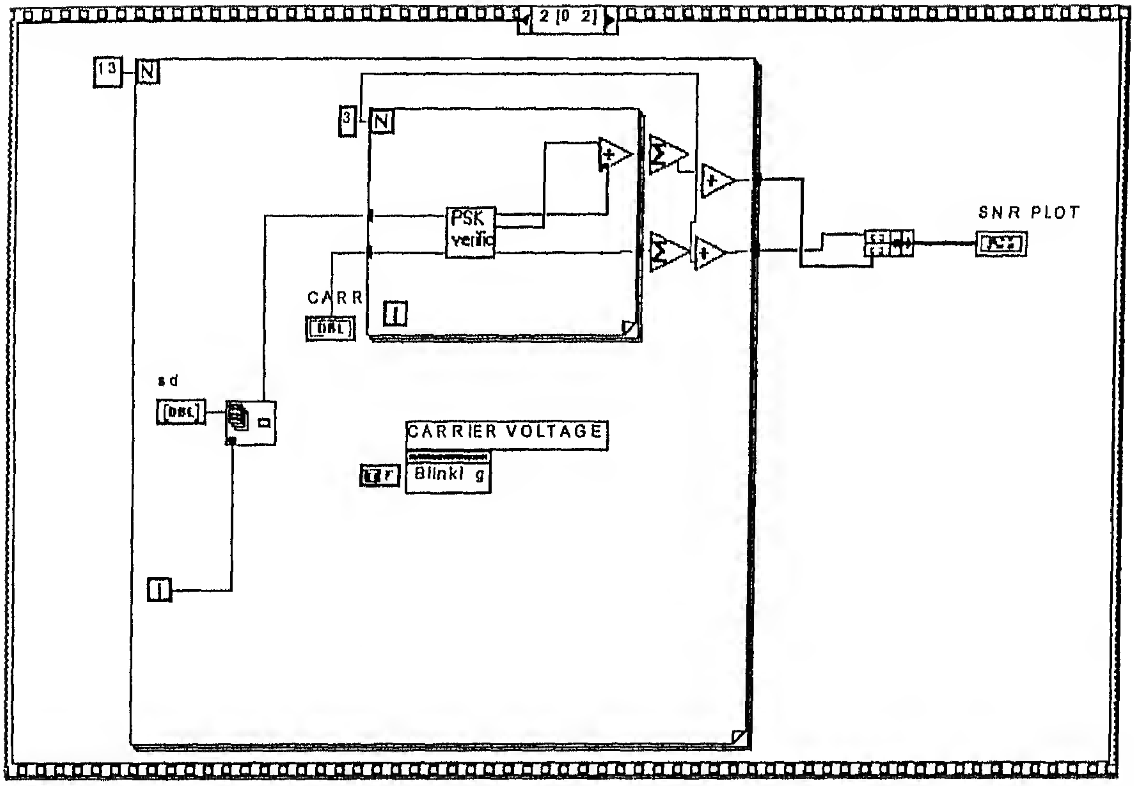


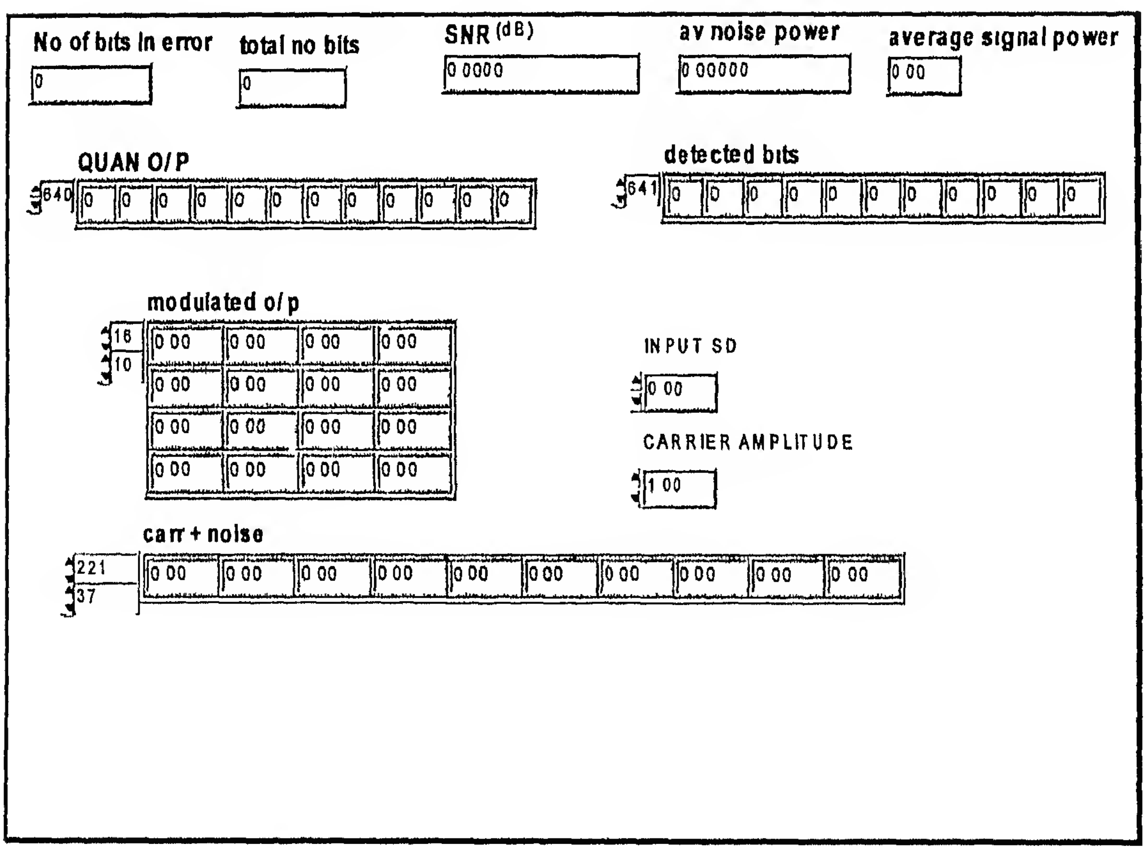
PRESET VI



PRESET VI




2 [0 2]
13
N
3
PSK
verific
CARR
DBL
s d
CARRIER VOLTAGE
Blinki g
SNR PLOT


No of bits in error
total no bits
SNR (dB)
av noise power
average signal power
QUAN O/P
detected bits
modulated o/p
INPUT SD
CARRIER AMPLITUDE
carr + noise

Reference

[1] Natıonal Instruments Corporatıon LabVIEW User Manual (1998)

[2] Natıonal Instruments Corporatıon Functıon and VI reference Manual (1998)

[3] Natıonal Instruments Corporatıon Data Acquısıtıon Basıcs Manual (1998)

[4] Lısa K Wells LabVIEW Students Edıtıon User s guıde (New Jersey Prentıce Hall 1995)

[5] Chughanı M L Samant A R Cerna M LabVIEW Sıgnal Processıng (New York, Prentıce Hall 1998)

[6] Jamel Rehman and Pıchlık Herbert LabVIEW Applıcatıons (New York Prentıce Hall 1998)

[7] Carl R Ryan Sımulatıon of Communıcatıon System Usıng Personal Computers ' IEEE Journal on selected Area s ın communıcatıon vol 6 no 1 pp 13 23 1988

[8] Kurt R Matıs and James W Modestıno "Interactıve Communıcatıon System Sımulatıon " IEEE Journal on selected area's ın communıcatıon vol 6 no 1 pp 24 33 1988

[9] Sebastıan Rıtz and Vojın Zıvojnovıc 'Hıgh Level Software Synthesıs for the Desıgn of Communıcatıon System,' IEEE Journal on selected areas ın communıcatıon vol 11 no 3 pp 348 358 1993

[10] Mıscha Schwartz 'Informatıon Transmıssıon Modulatıon and Noıse ' 3rd ed (New York McGraw Hıll Publıcatıon 1980)

[11] B P Lathı "Modern dıgıtal and Analogue Communıcatıon Systems ' 2nd ed (Oxford Publıcatıon 1989)

[12] Edward A Lee and Davıd G Messerschmıtt Dıgıtal Communıcatıon 2nd ed (London Kluwer Publıcatıon, 1994)

[13] Sımon Haykıns Communıcatıon Systems ' 3rd ed (New York Wıley Publıcatıons 1996)

[14] John G Proakıs ' Dıgıtal Communıcatıon, 3rd ed (New York McGraw Hıll Publıcatıon 1995)

[15] P K Chatterjee and J John 'Lıne Codes for Optıc Fıber Communıcatıon Systems IETE Journal of Educatıon vol 38 no 3&4 pp 197 211 1997

[16] John Bellamy "Wıley Serıes ın Telecom, 2 d ed (New York, Wıley Interscıence Publıcatıon 1985)

[17] K S Shanmugam, "Dıgıtal and Analogue Communıcatıon Systems (New York Wıley Publıcatıons 1985)

[18] Bernard Wıdrow & Samual D Stearns Adaptıve Sıgnal Processıng ' (New Jersey Prentıce Hall Publıcatıon 1985)

[19] W T Webb and L Hanzo Modern Quardrature Amplıtude Modulatıon ' (New York Pentech Publıcatıon, 1985)

[20] W D Gregg "Analogue and Dıgıtal Communıcatıon ' (New York Wıley Publıcatıon, 1977)

[21] Lucky Salz and Weldon, Prıncıpal of data Communıcatıon, (New York McGraw Hıll Publıcatıon 1968)

[22] Bernard Sklar ' Dıgıtal communıcatıons fundamentals and applıcatıons "

(New Jersey Prentice Hall Publication 1988)

[23] Dr K Feher Digital Communication Satelite\Earth Station Engineering (New Jersey Prentice Hall Publications 1987)

[24] Subbarayan Pasupathy Minimum Shift Keying IEEE Communication Magazine pp 14 22 1979

[25] R W Lucky Techniques of Adaptive Equalization of Digital Communication Systems Bell System Technical Journal pp 255 285 1966

[26] P Balaban K S Shanmugan and M C Jeruchim Simulation of communication systems (New York Plenum Press 1983)

A 130789

A130789